Scriptcase Overview

What is Scriptcase?

Scriptcase is a complete development tool, with a friendly web interface. A low-code environment that creates PHP applications high-speed and with quality was built to save time, reduce costs, gain productivity, and to integrate the development teams. Using Scriptcase, you can create many types of applications like **Grids, Charts, Forms, Calendars, and many other applications**. It also offers an IDE integrated for you to customize applications or even to create using only codes.

You can divide Scriptcase environment into 3 parts: **RAD (Rapid Application Development)**, **Scriptcase environment tools** and **Code editors**.

- **RAD (Rapid Application Development)**: Within Scriptcase, you can find some ready-to-use applications as well as complete system templates. You can see a sample of all applications accessing our website here.
- **Scriptcase environment tools**: You have available plenty of tools to assist your development within Scriptcase, such as Database Manager, SQL Builder, Data dictionary, CSV, XLS and Access converters, Documentation Creator, Themes Manager, Images Manager, Libraries Manager, Tasks Manager, and Key Shortcuts.
- **Code editors**: Besides the ready-to-use applications, templates, and tools, you can also customize or create your application using the editors, they work like an IDE integrate into Scriptcase. Within the editors, you can add your system rules integrating with web services, third-party libraries and components, APIs, and Scriptcase internal functions (macros). The editors are compatible with PHP, JavaScript, CSS, and HTML, and you can also use Emmet Shortcuts.

How does Scriptcase work?

Scriptcase works generating web applications in PHP and always relating the content with the most used databases on the market. Installing on a web server, Scriptcase can be accessed simultaneously by many developers through the browser, allowing remote and collaborative development. The source code is in PHP, and it works independently of the tool. Once you finish your project on Scriptcase, you can deploy it anywhere with a WEB server that allows you work with PHP. You do not need a license to run the generated applications. Scriptcase provides a “Production environment” where you can manage the libraries, connections, and deployed applications. Learn more about the deployment here.

Which source code is generated?

Scriptcase generates (PHP, JavaScript, and HTML) source code. The generated code is independent of the tool, and you can deploy it on a web server with PHP enabled. The PHP language is free (GNU), and you can use on any operating system (Windows, Linux, or Mac). PHP processes work on the Server.

What are the supported databases?

Scriptcase connects to the leading relational databases: Oracle, MS-SQL Server, DB2, MySQL, Informix, PostgreSQL, Access, InterBase, FireBird, Progress, and others ODBC or OLEDB patterns.

What are the requirements to run ScriptCase?

A webserver (Apache, IIS, Sambar, Abyss, PWS, Xitami) that is enabled to run PHP 8.1 Scripts and IonCube Loader.

Accessing Scriptcase

Access to Scriptcase must be done through a browser of your choice. The tool is compatible with the most used browsers.

To access, we must inform in the address bar of the browser: **Server IP:+port+Scriptcase folder name**

Example of access when performing local installation

- 127.0.0.1:8092/scriptcase
- localhost:8092/scriptcase

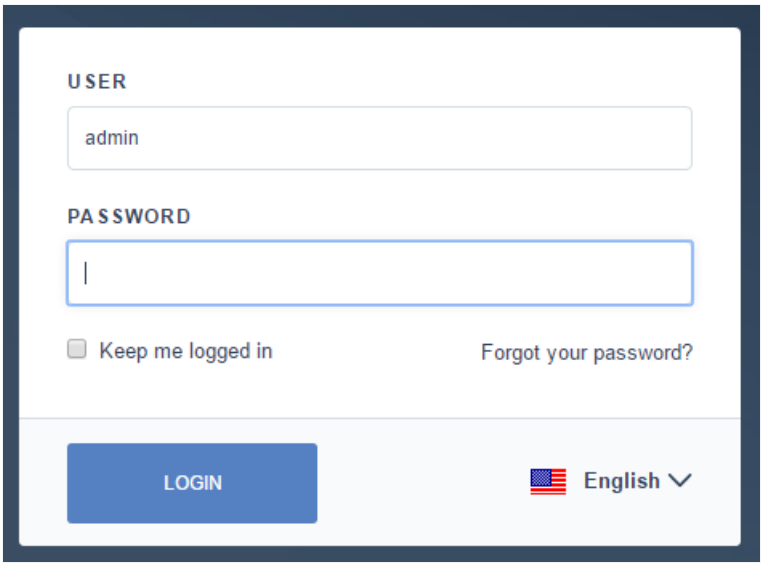
Example of access when performing installation on a server

- 180.204.163.144:8092/scriptcase/
- mysc.domain.net/scriptcase/

Compatible browsers

Chrome	Any version
Firefox	Version 4 or higher
Edge	Any version
Opera	Version 9 or higher
Safari	Version 5.2 or higher

Login screen



In addition to the **User** and **Password** fields, the login screen has some resources. **Keep me signed in**, **Forgot your password?** and the **Interface language selection**.

Keep me logged in

This option allows the browser to store the cookie and remain connected to your account when you close the browser tab where Scriptcase is open, and come back later without the need to log in again.

The cookie storage time is defined by accessing the menu Settings > Administration in the option **Time to expire the login**

cookie (days)

Forgot your password?

Allows recovery of the Scriptcase access password. For this option to work correctly, it is necessary to configure the SMTP in the tool settings.

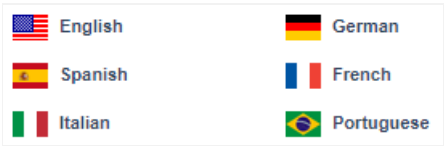
It is recommended that this configuration be performed on the first accesses.

Check here how to perform this configuration. Check here some SMTP options.

Interface Language Selection

Allows you to set the Scriptcase interface language.

Below are the available options.



The Portuguese language is only available in pt_br installers

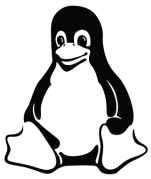
Related Links 🔗

Related video ▷

Scriptcase Installing Overview

- ScriptCase installer includes PHP 8.1, Apache 2.4.53, and SourceGuardian.
- The installation process is described according to the Operating System.

Installing ScriptCase is not complicated, select your operating system below, follow the tutorial, and in a few steps, you get to install and start using ScriptCase. If you need any assistance, please contact our technical support here.



Installer for Windows

The automatic installer quickly and simply configures the entire environment necessary to use Scriptcase. In addition to the tool itself, Apache and PHP will be configured, enabling essential libraries for the tool to function.

What will be installed

Scriptcase's automatic installer does not install any database.

- Apache Web Server **v2.4.53**.
- PHP **8.1.6** with extensions **MBSTRING**, **ZIP**, **BCMATH** and **GD** enabled.
- SourceGuardian **13.0.3**.
- Scriptcase

Installing Scriptcase with PHP 8.1 will not conflict with an existing installation of Scriptcase with PHP 7.0 or 7.3. The installer configures a different Apache service **(ApacheScriptcase9php81)** and port **(8092)** than the installer with PHP 7.0 or 7.3.

The installer is available on our website in x86 and x64 bit versions, download the installer according to your Windows architecture.

Compatibility Table

Scriptcase Automatic Installer is compatible with the following Windows versions

Windows	Windows Server
Windows 11	Windows Server 2022
Windows 10	Windows Server 2019
Windows 8.1	Windows Server 2016
Windows 8	Windows Server 2012 R2
Windows Vista SP2	Windows Server 2012
Windows 7 SP1	Windows server 2008 R2 SP1
	Windows server 2008 SP2

Installing Scriptcase with PHP 8.1 will not conflict with an existing Scriptcase installation using PHP 7.0 or PHP 7.3. The new Installer configures an Apache service **(ApacheScriptcase9php81)** with a different port **(8092)** than the installers with the PHP 7.0 or PHP 7.3.

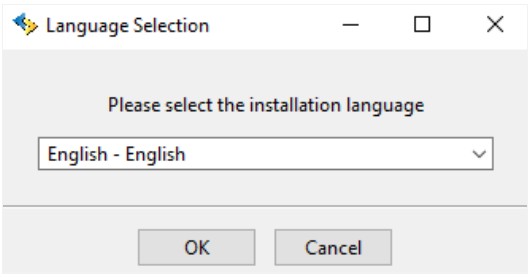
Check out the installation step-by-step below.

Scriptcase installation

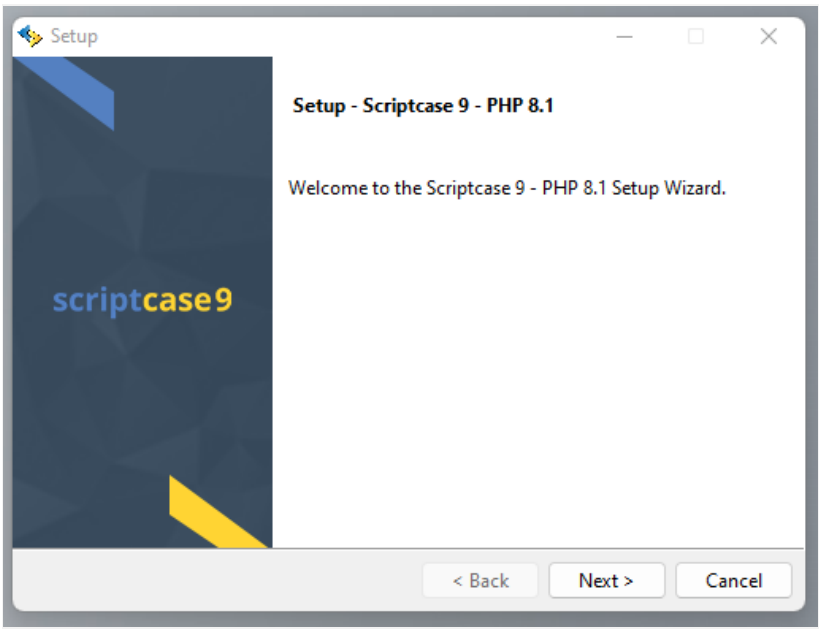
1 - Language Selection

You can select the language that will be displayed in the installation process. This step does not select the Scriptcase interface language, only the installer language.

Click **Ok** to continue.

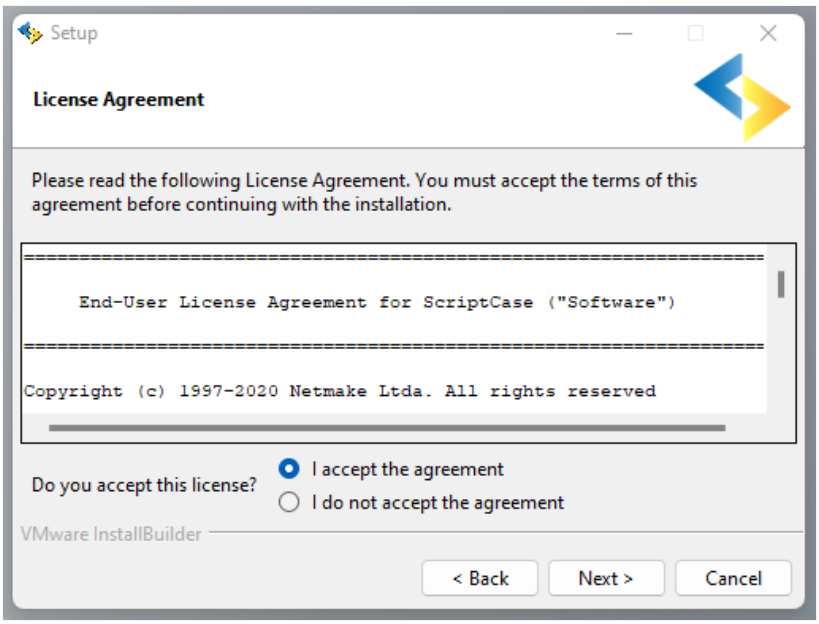


2 - Click “Next” to continue.



3 - License Agreement

Read and accept the terms of the license agreement

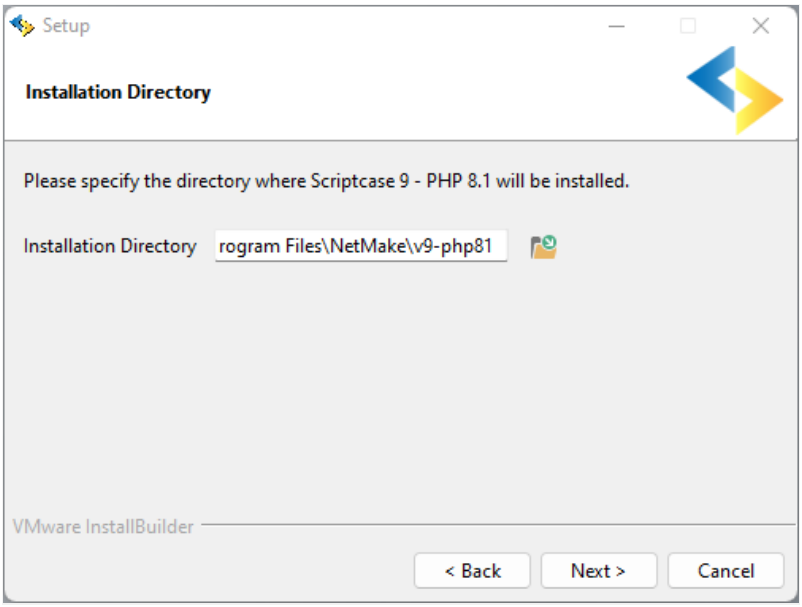


Click **Next** to continue.

4 - Installation Directory

The default installation path is **“C:\Program Files\NetMake\v9-php81”**. It is not recommended to make any changes to this path

unless it is necessary.



Installation directory

In this option, you can set the installation path.

: This button lets you select the installation folder or create a new one.

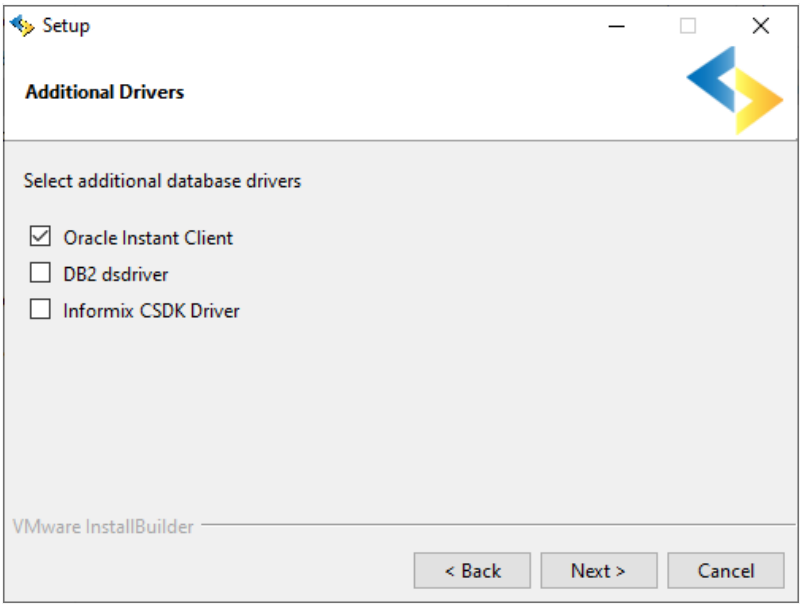
Click **Next** to continue.

5 - Drivers Adicionais

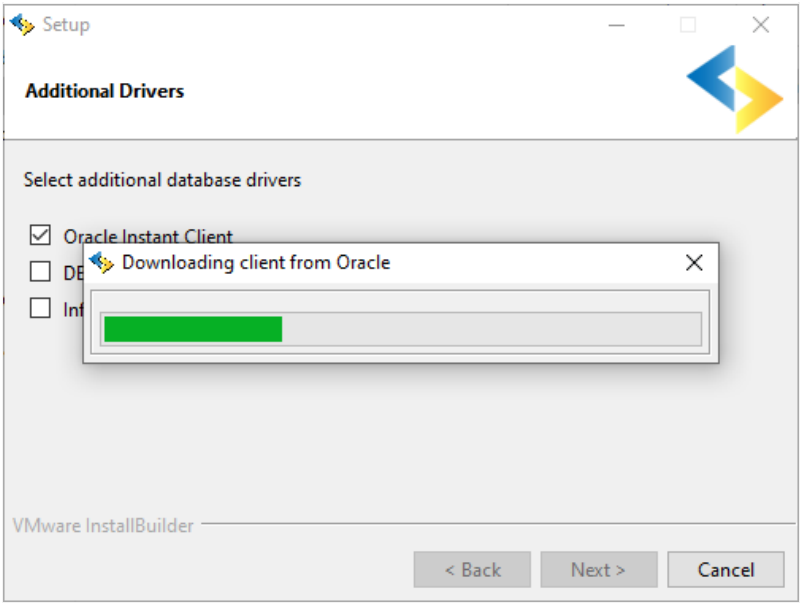
In this step, you can enable the drivers for the **Oracle**, **DB2** and/or **Informix** connections according to your needs. The other connections are automatically enabled when using our installer.

Choosing one of the drivers is optional, if you are not interested in enabling those drivers, just click**Next**.

This step is only available when using the x64 installer due to x86 architecture restrictions.

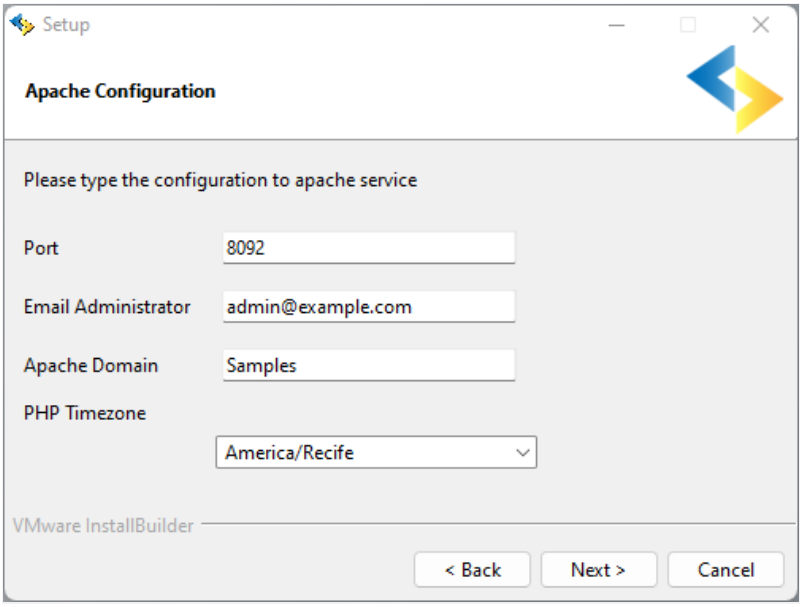


If you choose to configure any of the additional drivers, simply select the desired driver and click **Next**.



If you are using the x86 installer or have installed Scriptcase without selecting some of the drivers, check out our documentation on how to enable the Oracle, DB2 or Informix.

6 - Configure Apache.

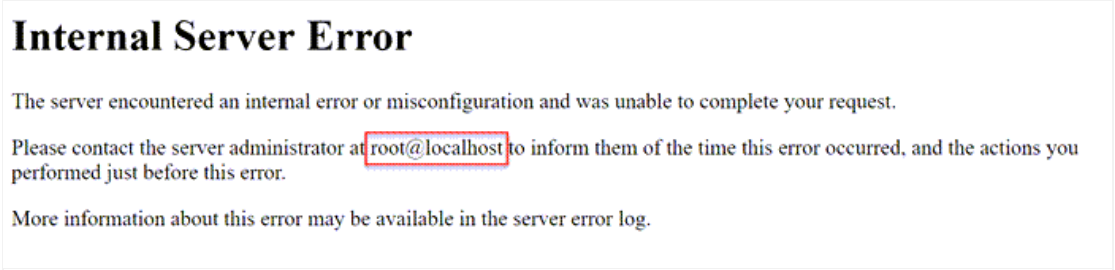


Port

Here will define in which port the Scriptcase Apache service will run on. We recommend using the value set in this installation. **(8092)**.

Email Administrator

You can configure Apache administrator email for better control of your environment. This setting serves as information when a server error occurs. A message will appear along with the administrator email to be contacted.



Apache Domain

The Apache domain must be set here. We do not recommend making any changes to the default value.

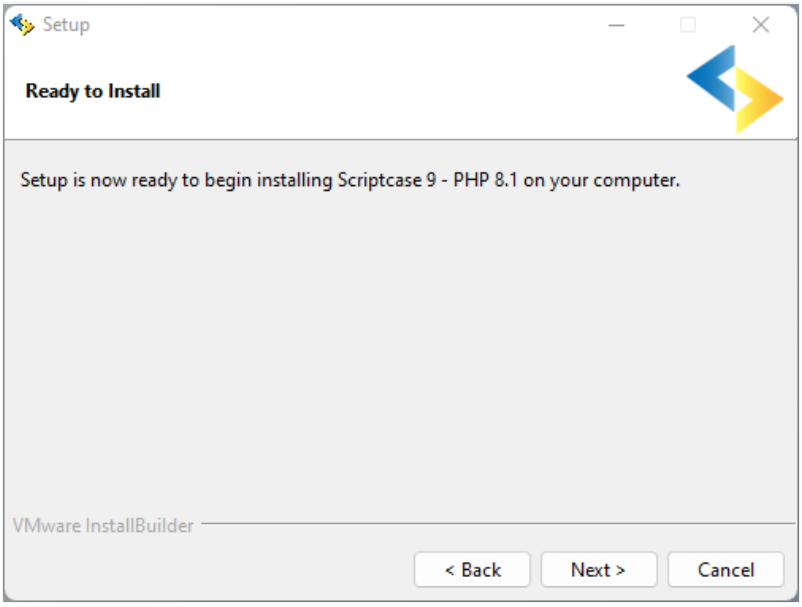
PHP TimeZone

In this option, you can set your PHP TimeZone. It is recommended to select the option corresponding to your time zone. You can check the list of available TimeZones here.

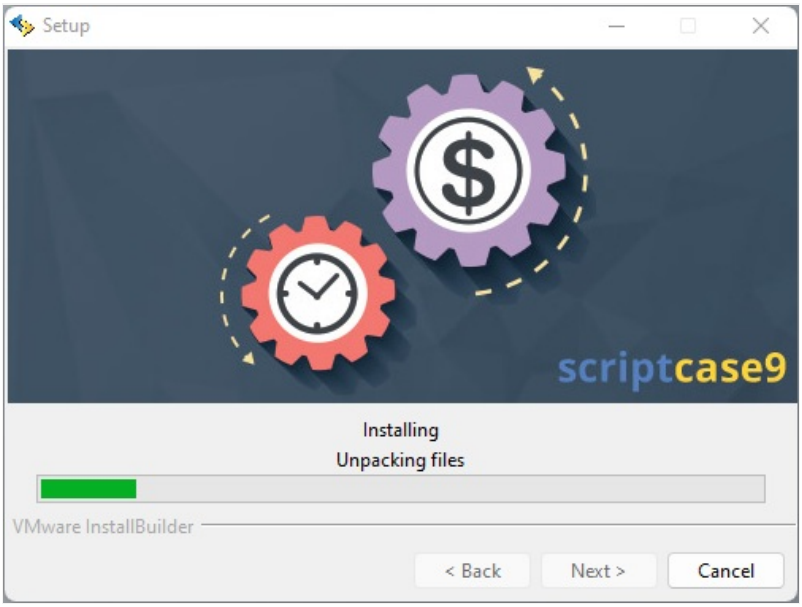
Click **Next** to continue.

7 - Ready to Install

Clicking “Next” will start the installation process.

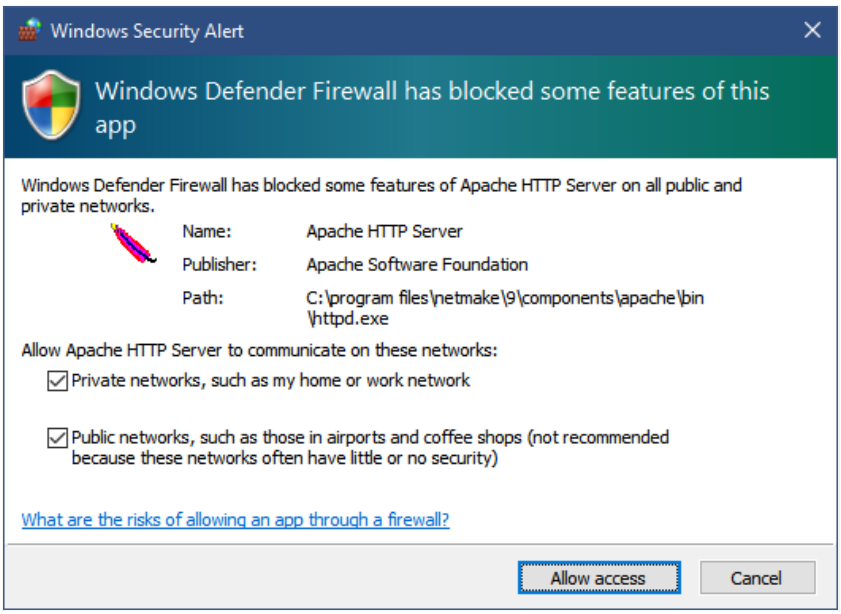


Wait until all components are installing and the environment is set up.



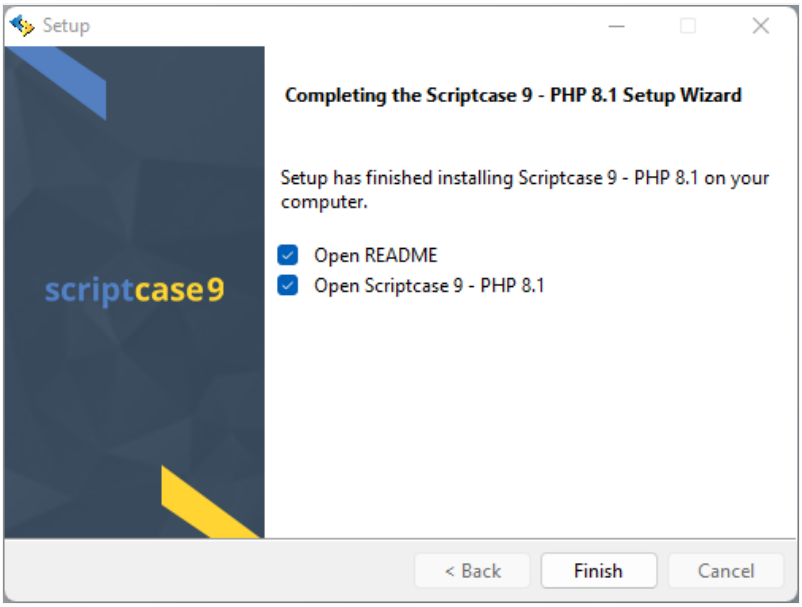
8 - Apache permission

Allow Apache to run on private and public networks (Firewall).



You must check both options to allow Scriptcase Apache to run on the operating system.

9 - Complete the installation



Open README

When you click Finish, you will be directed to a “Read Me” page with all the necessary initial information and useful links.

https://www.scriptcase.net/readme/

Open Scriptcase 9

Clicking **“Finish”**, will open Scriptcase in your default browser.

Access Scriptcase

1 - To access the Scriptcase you can use the shortcut created after installation or access using the browser.

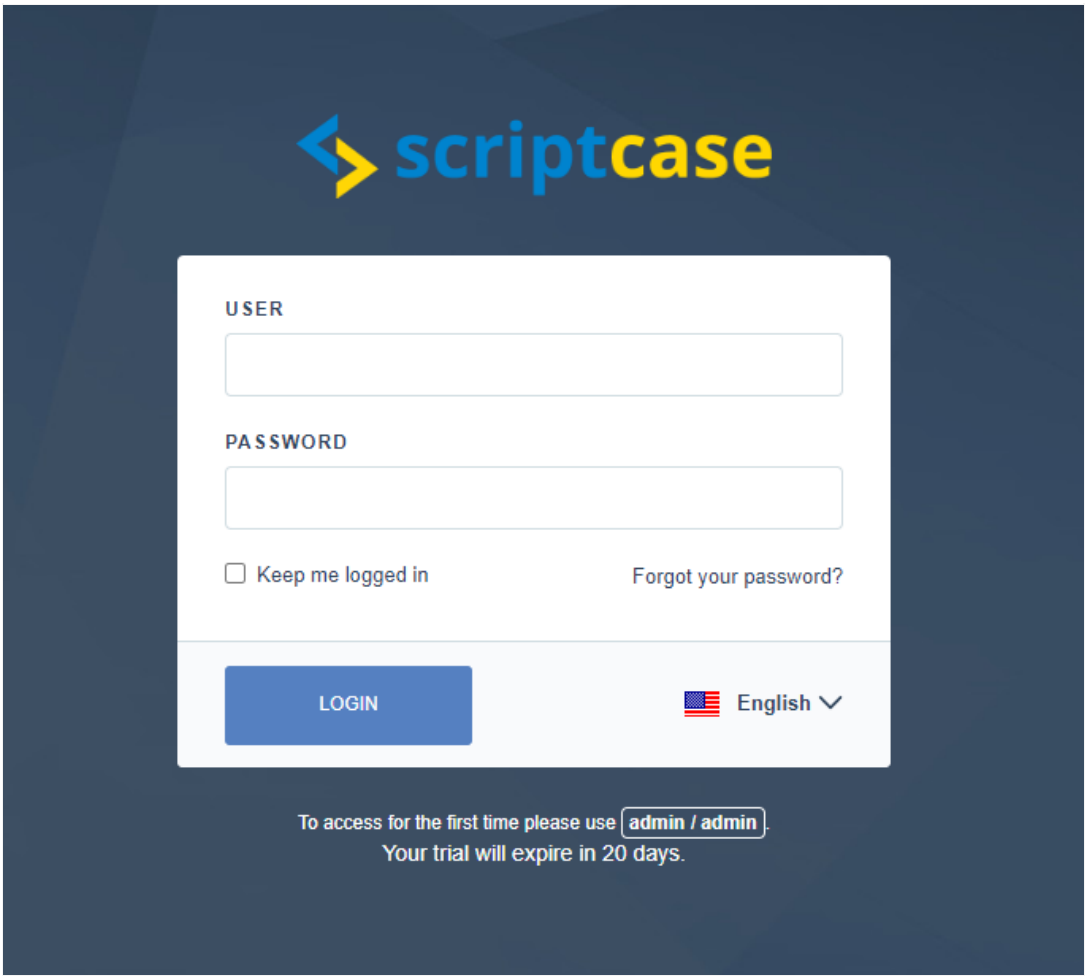
Example to access on the local install

- 127.0.0.1:8092/scriptcase
- localhost:8092/scriptcase

Example to access on the server install

- 180.204.163.144:8092/scriptcase/
- mysc.domain.net/scriptcase/

2 - After accessing the Login page, you must enter the username and password to proceed to your development environment.



After finishing the installation, you can use the version free of charge for a period of 20 days and you will have access to technical support via chat to help you during the evaluation. If you already have a serial, see how to register your scriptcase.

User

In this field, you must enter the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me logged in

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

This option makes it possible it is possible to recover the password to access Scriptcase. This feature is only available after configuring SMTP in Scriptcase settings. See how to set the SMTP

Installer for Linux

The Scriptcase installer installs **Apache, PHP and Scriptcase** on your operating system. **The installer does not include a database.**

Prerequisites

To download the Scriptcase Installer, go to our download page clicking here.

Listed below are the operating systems approved for Scriptcase use. **Check the list of supported distros:**

The Scriptcase installer is compatible with any distribution with its GLIBC 2.17 or higher

Distro	GLIBC	
RedHat 7	2.17	
RedHat 8	2.28	AlmaLinux, Rocky Linux, CentOS Stream
RedHat 9	2.28	AlmaLinux, Rocky Linux, CentOS Stream
Debian 8	2.19	MX, Q4OS, SparkyLinux, Zorin, Devuan, Deepin, Kali
Debian 9	2.24	MX, Q4OS, SparkyLinux, Zorin, Devuan, Deepin, Kali
Debian 10	2.28	MX, Q4OS, SparkyLinux, Zorin, Devuan, Deepin, Kali
Debian 11	2.31	MX, Q4OS, SparkyLinux, Zorin, Devuan, Deepin, Kali
Debian 12	2.36	MX, Q4OS, SparkyLinux, Zorin, Devuan, Deepin, Kali
Debian 13	2.37	MX, Q4OS, SparkyLinux, Zorin, Devuan, Deepin, Kali
Ubuntu 18.04	2.27	Elementary, Mint, Pop!_OS, Linux Lite, KDE neon
Ubuntu 20.04	2.31	Elementary, Mint, Pop!_OS, Linux Lite, KDE neon
Ubuntu 22.04	2.35	Elementary, Mint, Pop!_OS, Linux Lite, KDE neon
Ubuntu 23.04	2.37	Elementary, Mint, Pop!_OS, Linux Lite, KDE neon
SuSe 15	2.31	

Distro	GLIBC	
openSuSe leap (15)	2.31	
openSuSe tumbleweed	2.38	
Fedora 37	2.36	
Fedora 38	2.37	
Fedora 39	2.38	
AWS 1	2.17	Amazon Linux
AWS 2	2.26	Amazon Linux
AWS 3	2.34	Amazon Linux
ArchLinux base-20230910.0.177821	2.38	EndeavourOS, Manjaro, Garuda, ArcoLinux

NOTE: Before proceeding with the installation, it is recommended not to have any other web environments installed (such as LAMP, Zend Server, etc) on your operating system. **It is recommended to have some basic knowledge of Linux/Unix environment to use the Terminal (Shell).**

This documentation will describe Scriptcase installation in 3 simple steps:

1 - Run Scriptcase Installer.
2 - Choose installer settings.
3 - Run Scriptcase.

Running Scriptcase Installer

1 - Browse to the directory where the installer is located.

Before we can run the Scriptcase installer, we need to access the folder where the installer file is located using the `cd` command.

Example: `sudo cd /home/your_user/Downloads`

NOTE: Scriptcase installer requires administrative permissions to make changes to your machine. We will have to use the `sudo` command to run the installer.

2 - Grant permission to the installer file and run it.

For you to run the `chmod` command that changes the permissions of a file or directory, you must also run the `sudo` command.

Example: `sudo chmod +x scriptcase_linux_installer_en_us_v9.9.021-php81.run`

The `chmod + x` command is used to make the installer **.run** file executable.

3 - Run the Installer file.

Example: `sudo ./scriptcase_linux_installer_en_us_v9.9.021-php81.run`

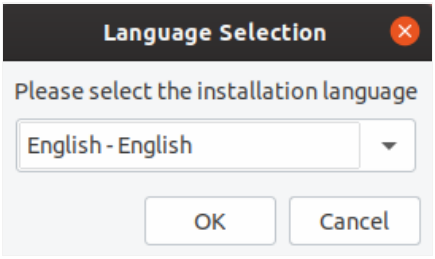
Choosing the Installer settings

NOTE: Installing Scriptcase with PHP 8.1 will not conflict with an existing Scriptcase installation with PHP 7.0 or 7.3. The Installer configures an Apache service **(apachesc9php81)** with a different port **(8092)** than the installer with PHP 7.0. or 7.3

Will be automatically installed on the operating system:

- Apache Web Server **v2.4.53**.
- PHP **8.1** with extensions **MBSTRING**, **ZIP**, **BCMATH** and **GD** enabled.
- SourceGuardian: **13.0.3**.
- Scriptcase.

1 - Choose the installation language.

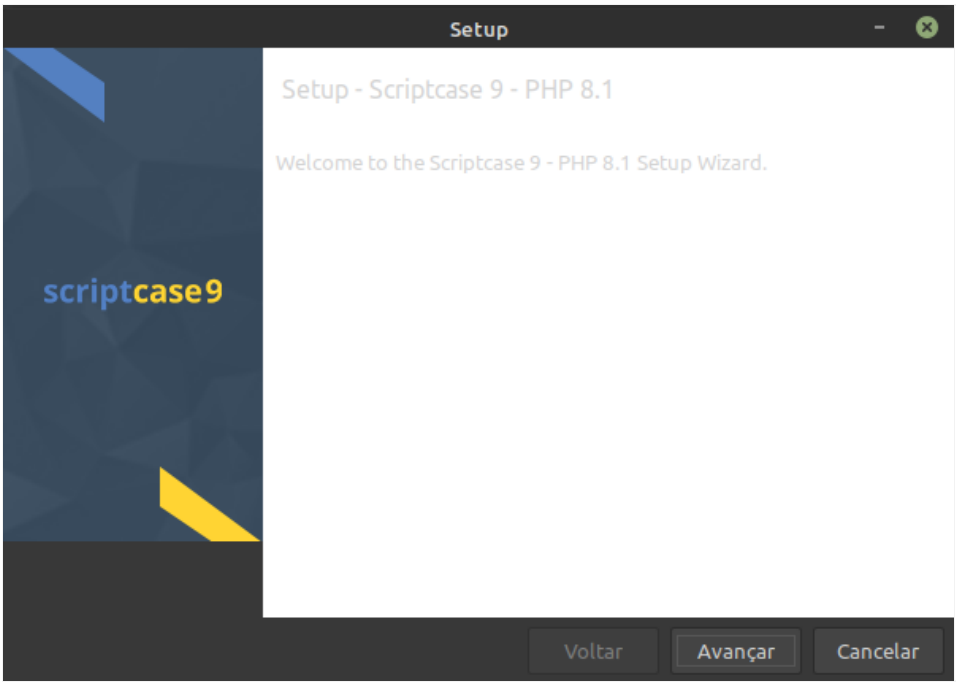


Language

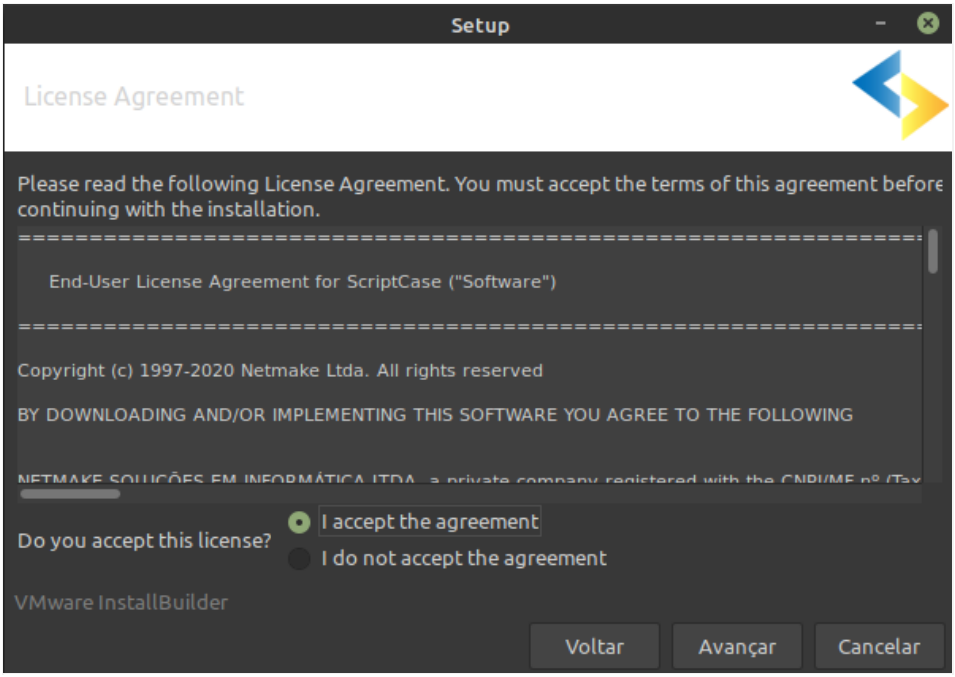
You can select the language that will be displayed in the installation process. This step does not select the Scriptcase interface language, only the installer language.

Click **Ok** to continue

2 - Click “Next” to continue.



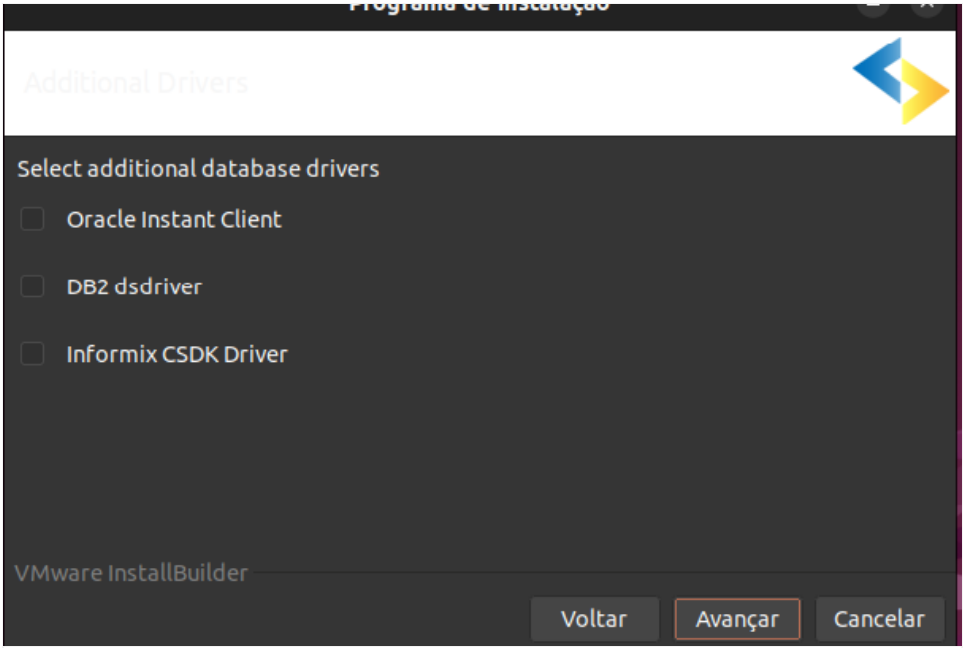
3 - Read and accept the terms of the license agreement.



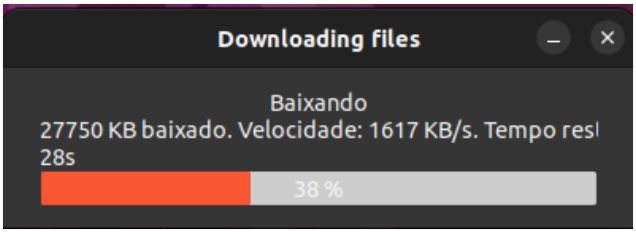
Click **Next** to continue.

4 - Select drivers to install.

The installer will ask if you want to add some additional drivers to the installation process: **Oracle instant Client, DB2 dsdriver and Informix CSDK Driver.**



When you click next, the installer will download the chosen drivers

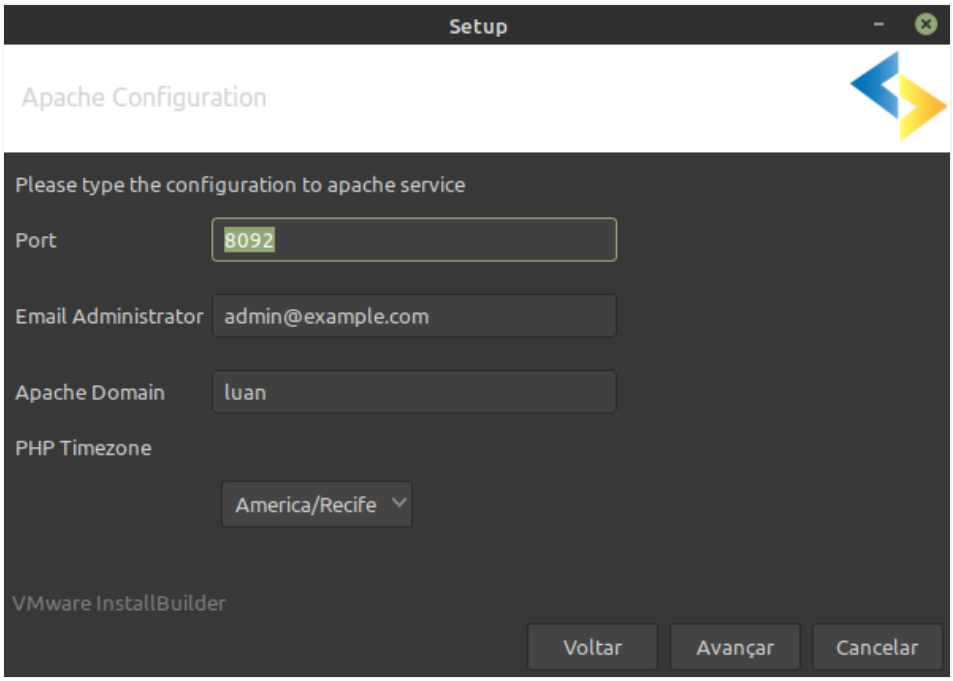


If you choose not to include any of these drivers at installation time, you can add them manually. The path where you must add the drives follows the following folder structure: **/opt/Scriptcase/v9-php81/components/drivers/driver_folder**

The following table shows the additional drivers, their respective directories and links to downloads

Driver	Path	Driver - link x64
OCI8/PDO_OCI	*/opt/Scriptcase/v9-php81/components/drivers/instantclient*	Download
PDO_INFORMIX	*/opt/Scriptcase/v9-php81/components/drivers/IBM/informix*	Download
PDO_IBM, IBM_DB2	*/opt/Scriptcase/v9-php81/components/drivers/IBM/Db2/dsdriver/*	Download

5 - Configure Scriptcase Apache.

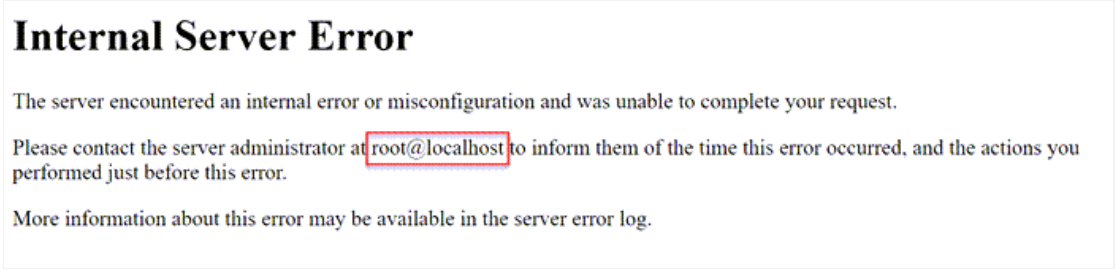


Port

Here will define in which port the Scriptcase Apache service will run on. We recommend using the value set in this installation **(8092)**.

Admin Email

You can configure Apache administrator email for better control of your environment. This setting serves as information when a server error occurs. A message will appear along with the administrator email to be contacted.



Apache Domain

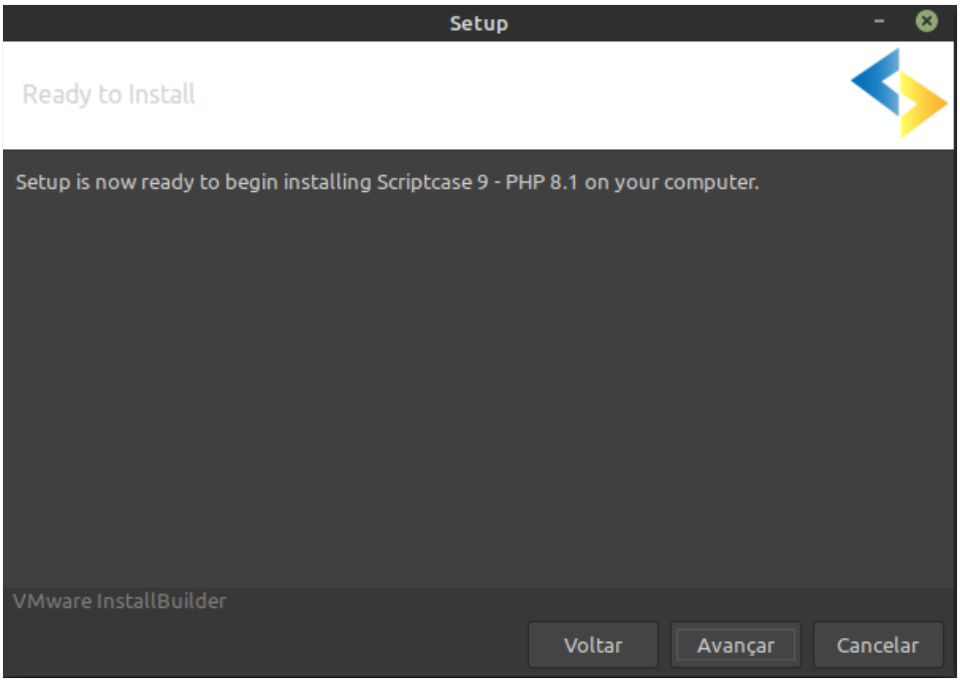
The Apache domain must be set here. We do not recommend making any changes to the default value.

PHP TimeZone

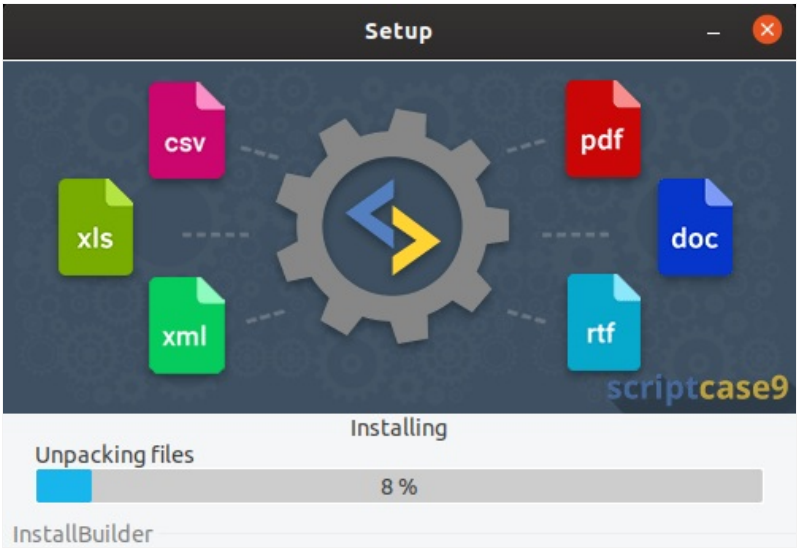
In this option, you can set your PHP TimeZone. It is recommended to select the option corresponding to your time zone. You can check the list of available TimeZones here.

Click **Next** to continue.

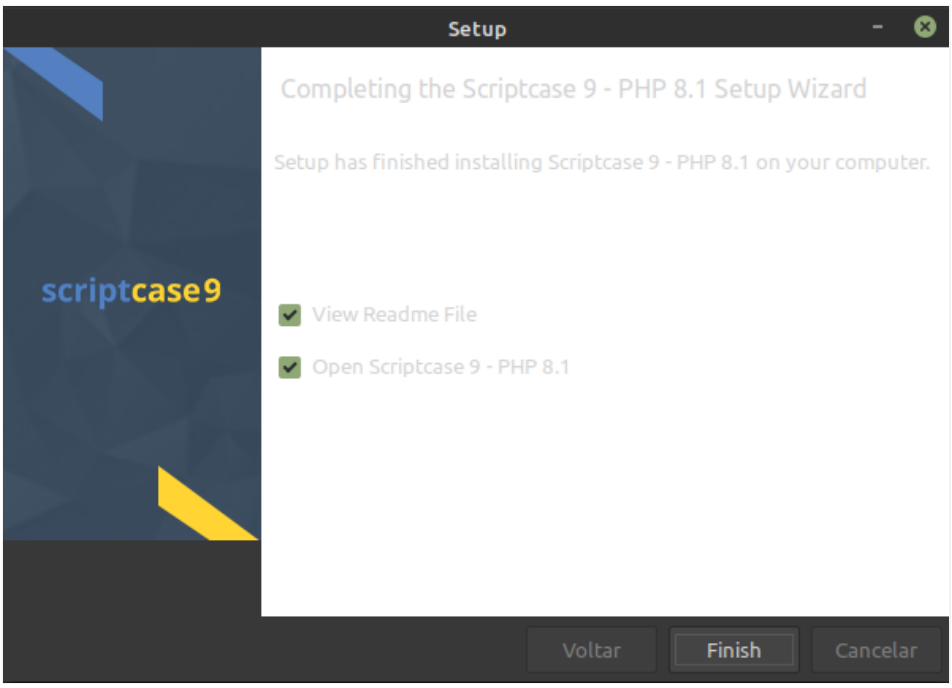
6 - Clicking “Next” will start the installation process.



Wait until all components are installing and the environment is set up.

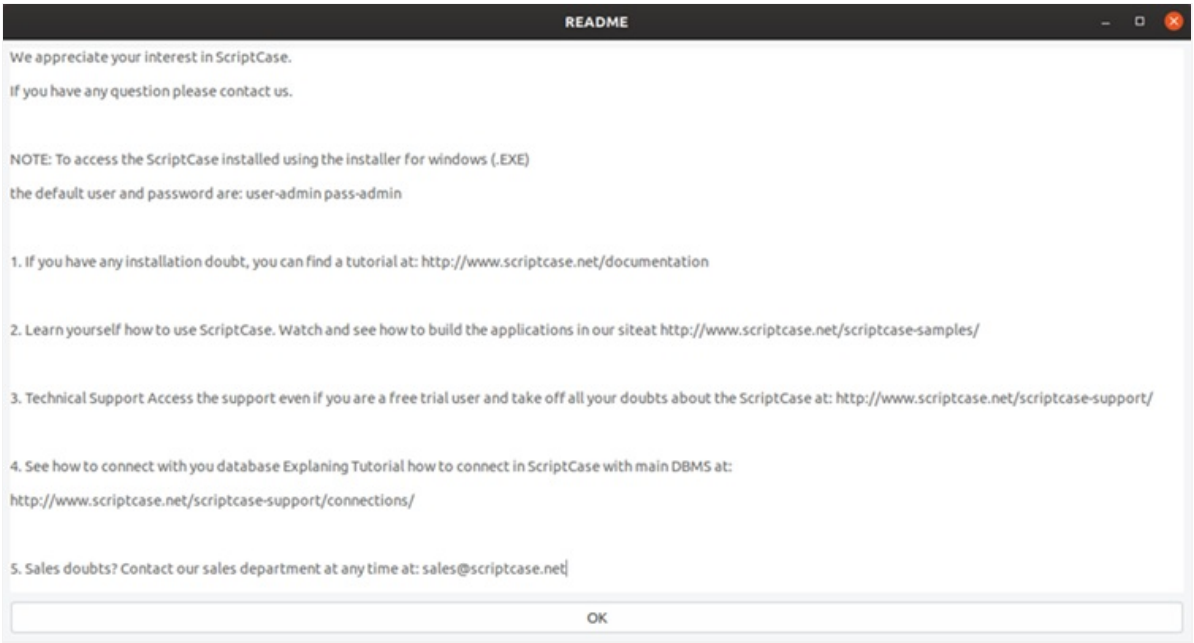


7 - Complete the installation.



View the “Read Me” file

Clicking finish you will be redirected to the **“Read Me”** page with all the necessary initial information and useful links.



Open Scriptcase 9

Clicking **“Finish”**, will open Scriptcase in your default browser.

Access the Scriptcase

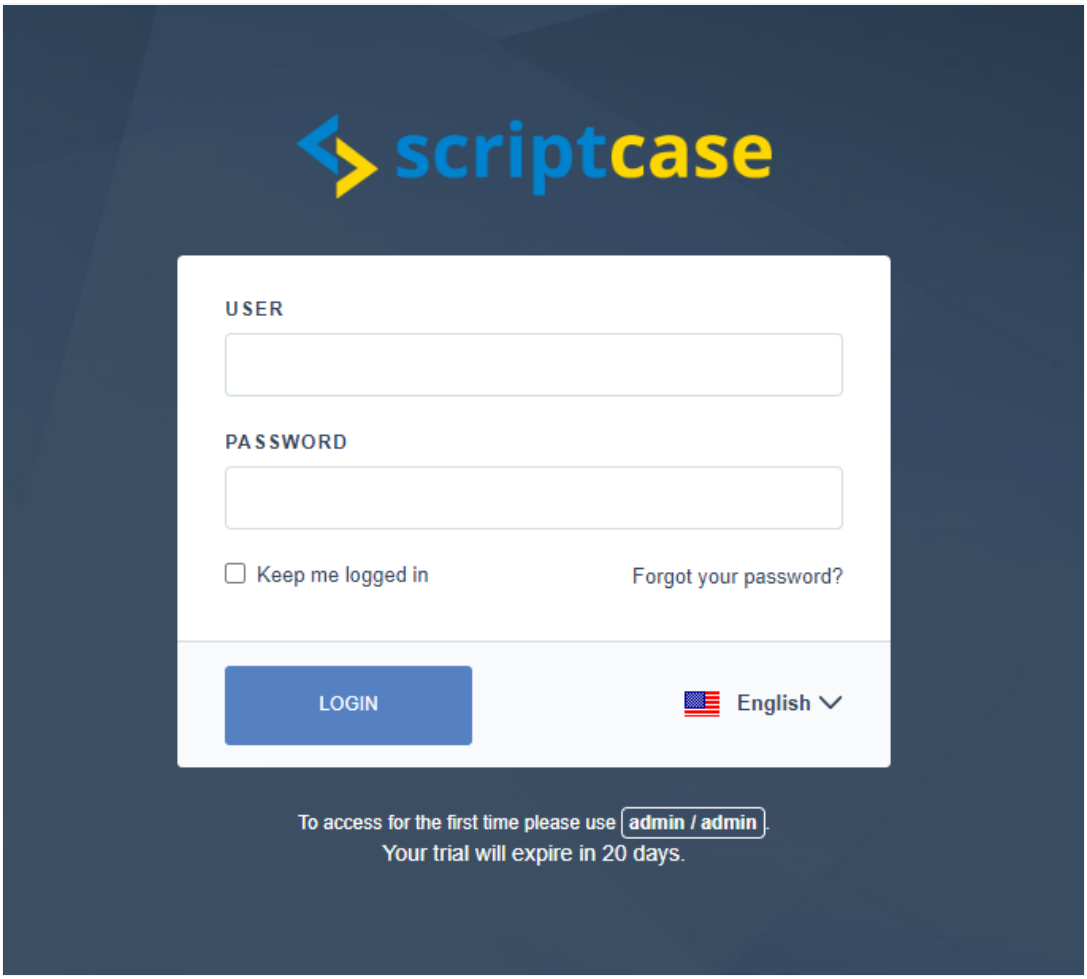
1 - To access your Scriptcase, you need to open your browser and enter the local IP or domain of the ported server you selected in the Apache setup.

Example 1: `http://127.0.0.1:8092/scriptcase`

Example 2: `https://yourdomain.com:8092/scriptcase`

2 - Login to Scriptcase

After accessing the Login page, you must enter the username and password to proceed to your development environment.



User

In this field, you must enter the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me logged in

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

IMPORTANT: Password recovery will only be possible if you have configured your SMTP information. To access our documentation showing how to perform this configuration click here.

By clicking on this option, it is possible to recover the password of the informed user. An email will be sent to the user so they can access the recovery steps.

Need to register your Scriptcase?

You can access our knowledge base to understand all the steps for registering a Scriptcase license.

Installer for macOS

The Scriptcase installer installs **Apache, PHP and Scriptcase** on your operating system.

The Scriptcase installer does not include a database.

Prerequisites

To download the Scriptcase Installer, go to our download page clicking here.

IMPORTANT: The Scriptcase installer is a **.app** file. To run the file, you will need to press and hold the **cntrl** key, mouse/touchpad right-click, and then click **Open** to authenticate as an administrator and install Scriptcase.

Listed below are the macOS versions approved for Scriptcase use.

Compatibility Table

Scriptcase Automatic Installer is compatible with the following macOS versions

Version	Codename
macOS 15.0	“Sequoia”
macOS 14.2	“Sonoma”
macOS 13.0	“Ventura”
macOS 12.0	“Monterey”
macOS 11.0	“Big Sur”
macOS 10.15	“Catalina”
macOS v10.14	“Mojave”
macOS v10.13	“High Sierra”
macOS v10.12	“Sierra”
OS X v10.11	“El Capitan”

Before proceeding with the installation, it is recommended not to have any other web environments installed (such as MAMP, Zend Server, etc) on your operating system.

Choosing settings in the Installer

Installing Scriptcase with PHP 8.1 will not conflict with an existing Scriptcase installation using PHP 7.0 or PHP 7.3. The new Installer configures an Apache service **(ApacheScriptcase9php81)** with a different port **(8092)** than the installers with the PHP 7.0 or PHP 7.3.

Will be automatically installed on the operating system:

- Apache Web Server **v2.4.53**.
- PHP **8.1** with extensions **MBSTRING**, **ZIP**, **BCMATH** and **GD** enabled.

- SourceGuardian **13.0.3**.
- Scriptcase **9**.

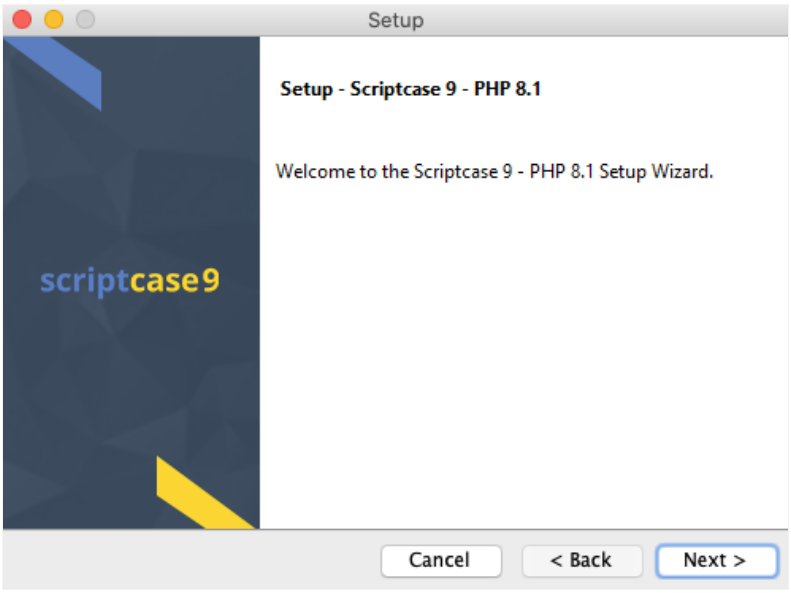
1 - Choose the installation language.

Language

You can select the language that will be displayed in the installation process. This step does not select the Scriptcase interface language, only the installer language.

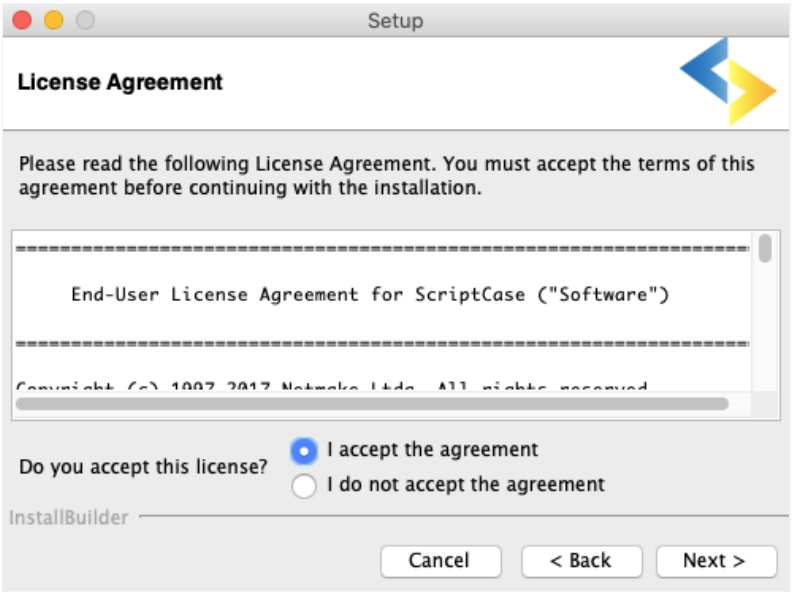
Click **Ok** to continue

2 - Click “Next” to continue.



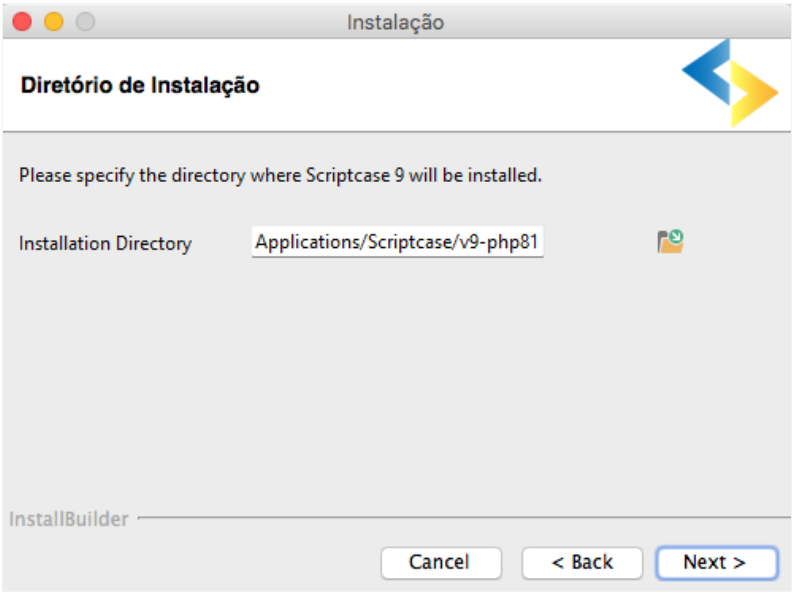
3 - Read and mark I accept the agreement.

Click **Next** to continue.



4 - Enter the installation directory.

The default installation path is **"/Applications/Scriptcase/v9-php81"**. It is not recommended to make any changes to this path unless it is necessary.



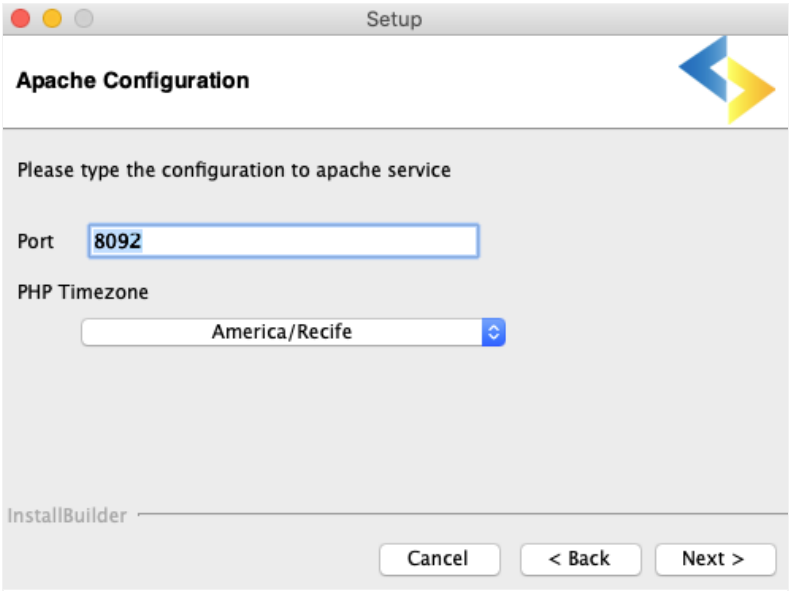
Installation directory

In this option, you can set the installation path.

: This button lets you select the installation folder or create a new one.

Click **Next** to continue.

5 - Configure Scriptcase Apache.



Port

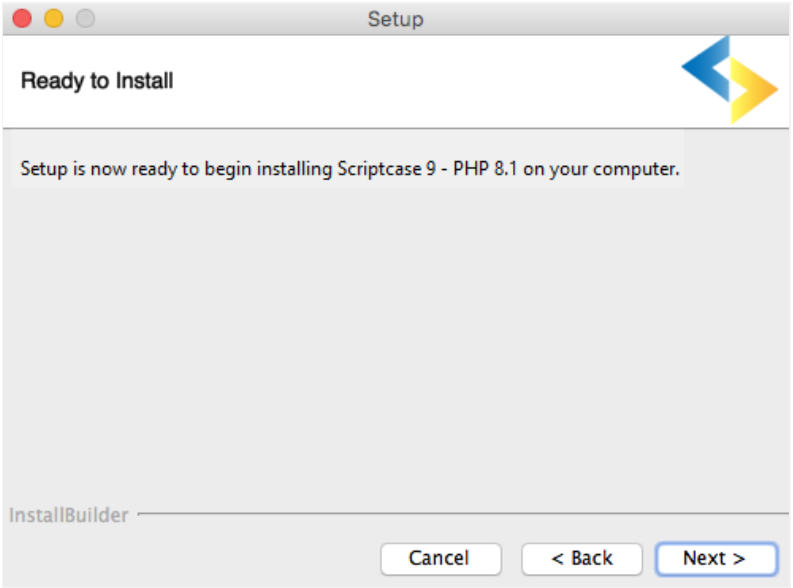
Here will define in which port the Scriptcase Apache service will run on. We recommend using the value set in this installation **(8092)**.

PHP TimeZone

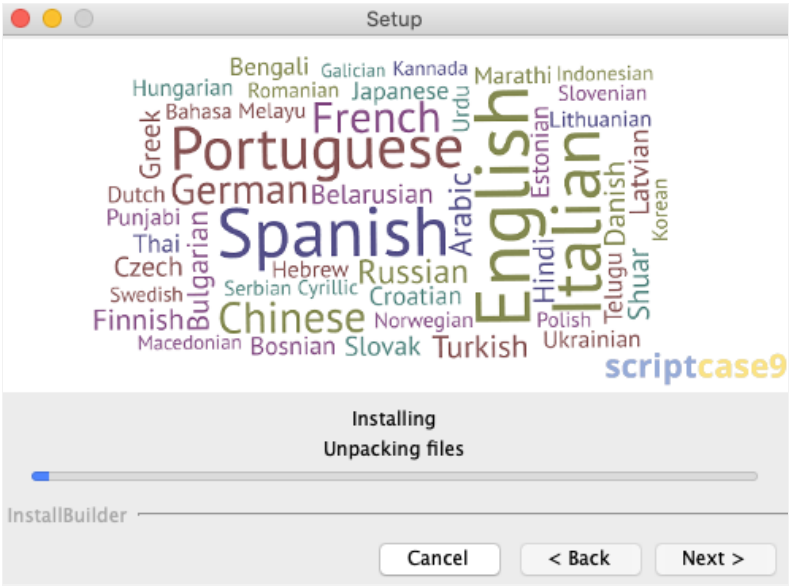
In this option, you can set your PHP TimeZone. It is recommended to select the option corresponding to your time zone. You can check the list of available TimeZones here.

Click **Next** to continue.

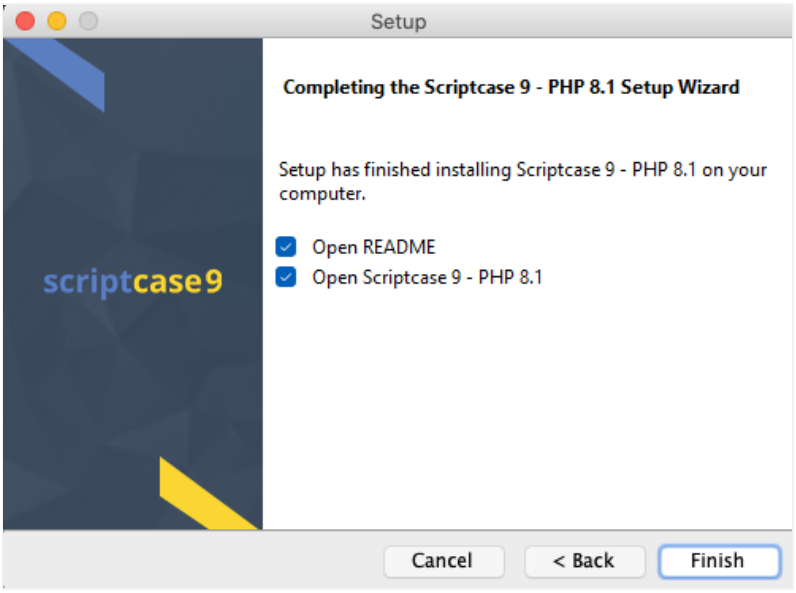
6 - Clicking "Next" will start the installation process.



Wait until all components are installing and the environment is set up.



7 - Complete the installation.



Open README

When you click Finish, you will be directed to a “Read Me” page with all the necessary initial information and useful links.

https://www.scriptcase.net/readme/

Open Scriptcase 9

Clicking finish **“Finish”** will open Scriptcase in your default browser.

Access Scriptcase

1 - To access the Scriptcase you can use the shortcut created after installation or access using the browser.

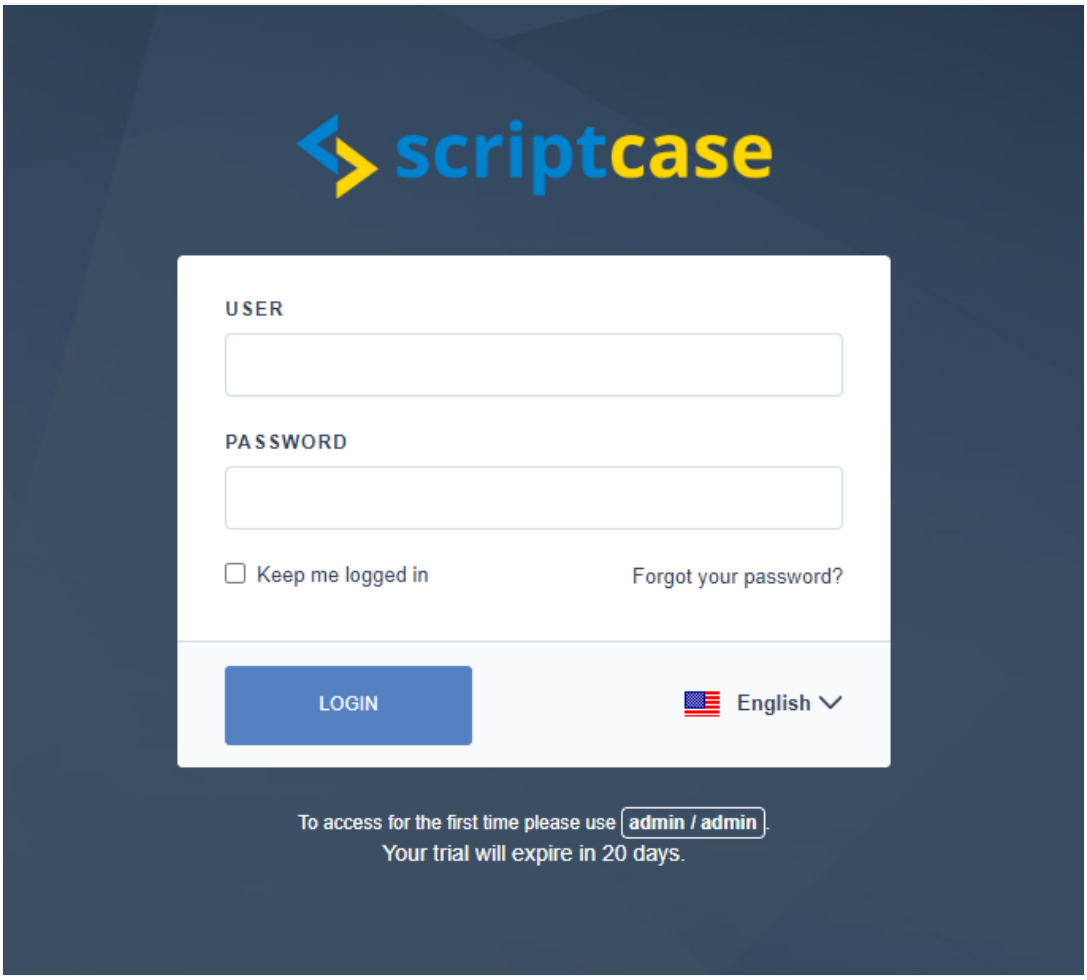
Example to access on the local install

- 127.0.0.1:8092/scriptcase
- localhost:8092/scriptcase

Example to access on the server install

- 180.204.163.144:8092/scriptcase/
- mysc.domain.net/scriptcase/

2 - After accessing the Login page, you must enter the username and password to proceed to your development environment.



After finishing the installation, you can use the version free of charge for a period of 20 days and you will have access to technical support via chat to help you during the evaluation. If you already have a serial, see how to register your scriptcase.

User

In this field, you must enter the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me logged in

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

This option makes it possible it is possible to recover the password to access Scriptcase. This feature is only available after configuring SMTP in Scriptcase settings. See how to set the SMTP

Installing PHP 8.1 - Windows

This documentation will walk you through the steps necessary to install PHP 8.1, Apache 2.4, and Scriptcase manually. **When performing this type of installation, you are responsible for configuring your entire environment as well as database extensions.**

You can install the Scriptcase with PHP 7.3 or PHP 7.0: See how to install with PHP 7.3 See how to install with PHP 7.0

Listed below are all the supported operating systems as well as the libraries that are required to function on a particular system.

Compatibility Table

Scriptcase Automatic Installer is compatible with the following Windows versions

Windows	Windows Server
Windows 11	Windows Server 2022
Windows 10	Windows Server 2019
Windows 8.1	Windows Server 2016
Windows 8	Windows Server 2012 R2
Windows Vista SP2	Windows Server 2012
Windows 7 SP1	Windows server 2008 R2 SP1
	Windows server 2008 SP2

For **Windows 7 SP1**, you must verify if the system has the following items installed: Visual C++ Redistributable Packages for Visual Studio (2008, 2010, 2012, 2013, 2015) **x86** and **x64**.

Prerequisites

If you already have a Web Server with Apache and PHP 8.1 configured, just go to How to Enable SourceGuardia Loader and proceed to manual installation.

To proceed with the installation, you need to download some files.

Check below for the files you will need:

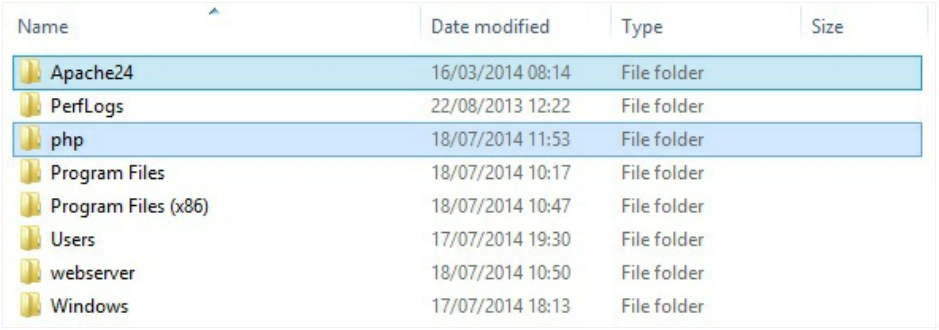
- PHP **8.1 NTS (x86 or x64)**: Click Here
- Apache **2.4 (x86 or x64)**: Click Here
- mod_Fcgi **(x86 or x64)**: Click Here
- SourceGuardian Loader 13.0.3 for Windows **(x86 or x64)**: Click Here
- Scriptcase **(.zip)**: Click Here

NOTE: Before proceeding with the installation, it is recommended not to have any other web environments installed (such as WampServer, XAMPP, Zend Server, etc.) on your operating system.

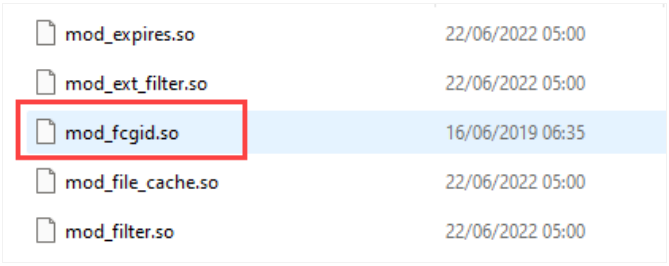
Below are the actions required after downloading the items specified above::

1 - Unzip the **PHP** and **Apache** files in separate folders on your operating system's C: drive.

2 - Rename the PHP folder to **php**. The result should be like this:



3 - Unzip the **mod_Fcgi** file, copy the **mod_fcgid.so** file to the `C:\Apache24\modules` folder.



Configure the Apache environment

Listed below are the steps required to configure apache manually. **Follow the steps correctly to avoid problems installing and starting the service.**

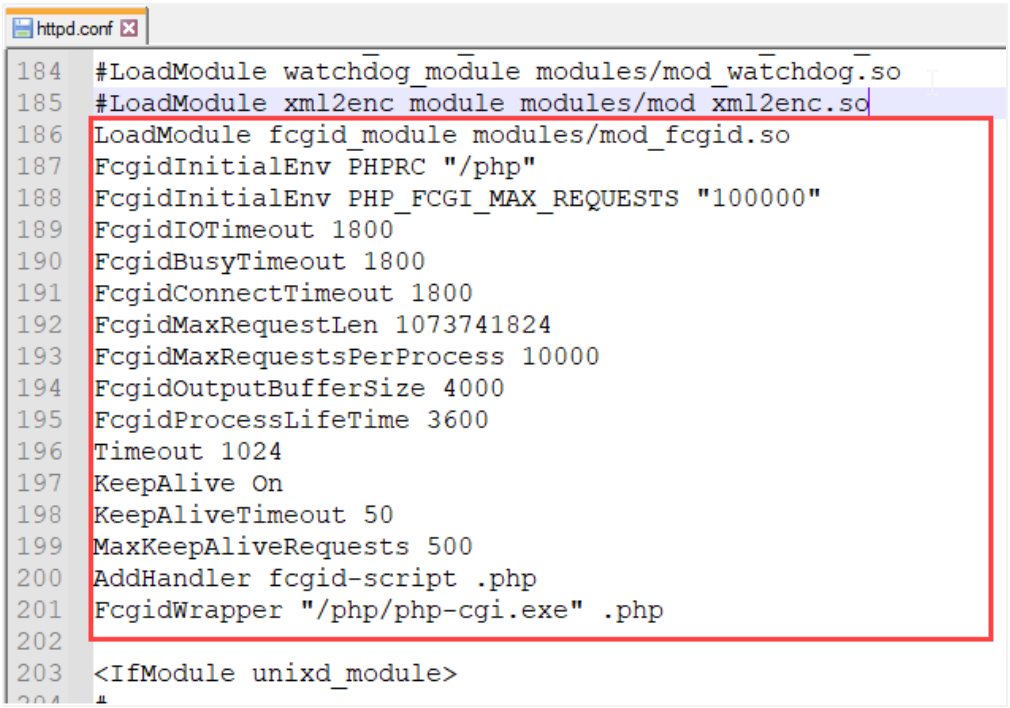
NOTE: Since PHP and Apache folders are at the same directory level **(C:)** You do not need to enter the root folder in directives that require you to enter isolated paths or points to a specific file.

1 - Edit the **httpd.conf** file located in the `C:\Apache24\conf\` folder using a text editor in Administrator mode, so that any changes you make are saved.

2 - Add the content below after the `#LoadModule xml2enc_module modules/mod_xml2enc.so` line:

```
LoadModule fcgid_module modules/mod_fcgid.so
FcgidInitialEnv PHPRC "/php"
FcgidInitialEnv PHP_FCGI_MAX_REQUESTS "100000"
FcgidIOTimeout 1800
FcgidBusyTimeout 1800
FcgidConnectTimeout 1800
FcgidMaxRequestLen 1073741824
FcgidMaxRequestsPerProcess 10000
FcgidOutputBufferSize 4000
FcgidProcessLifeTime 3600
Timeout 1024
KeepAlive On
KeepAliveTimeout 50
MaxKeepAliveRequests 500
AddHandler fcgid-script .php
FcgidWrapper "/php/php-cgi.exe" .php
```

Example:



WARNING: This setting will only work if you have copied the **mod_fcgid.so** file to the `C:\Apache24\modules` folder as previously instructed.

3 - Search for `#ServerName www.example.com:80` and change this line to `ServerName localhost:80` .

In this tutorial apache is configured with port 80. To define another port it is necessary to change the value informed in ServerName, for example (localhost:8092). Also, you will need to change the `Listen` , for example (Listen 8092). Before changing you must make sure that the port is not being used by another service

4 - Change the contents of the `DirectoryIndex` directive by adding the following contents to the directive line:: `index.php index.phtml` .

Example:

```
<IfModule dir_module>
    DirectoryIndex index.html index.php index.phtml
</IfModule>
```

```
#
<IfModule dir_module>
    DirectoryIndex index.html index.php index.phtml
</IfModule>

```

5 - Set the **${SRVROOT}** variable with your Apache folder path:

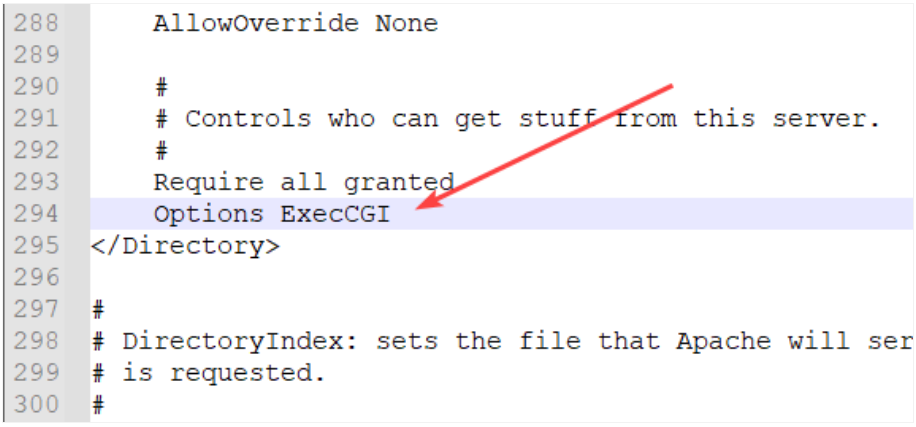
Search for the **Define SRVROOT** and edit it as follows:

`Define SRVROOT "/Apache24"`

6 - Add the **Options ExecCGI** command below the **Require all granted** line in the `<Directory "${SRVROOT}/htdocs">` directive.

```
    #
    Require all granted
 Options ExecCGI
</Directory>
```

Example:



7 - Search for the `<IfModule mime_module>` directive and add the content below **above** this directive:

```
<Directory "/php">
    #
    AllowOverride None
    Options None
    Require all granted
    #
</Directory>
```

Example:

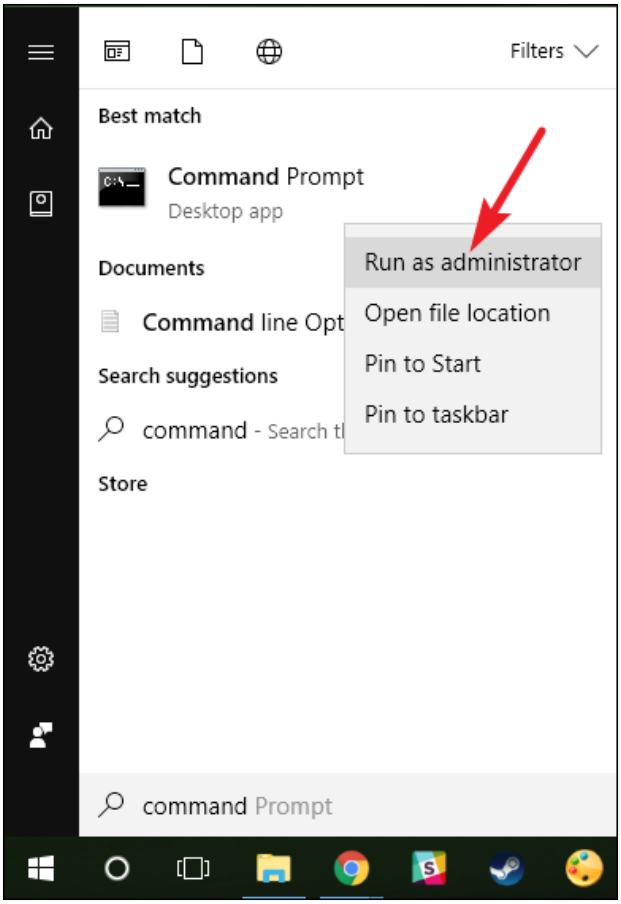
```
413     RequestHeader unset Proxy early
414 </IfModule>
415
416 <Directory "/php">
417     #
418     AllowOverride None
419     Options None
420     Require all granted
421     #
422 </Directory>
423
424 <IfModule mime_module>
425     #
426     # TypesConfig points to the file cc
427     # filename extension to MIME-type.
```

After performing all these procedures, save the `httpd.conf` file to proceed with Apache installation and startup and PHP configuration.

Install and start Apache

Now that you have set up your apache service, all you have to do is install and start it. Below are the steps that must be followed to successfully complete this process.

1 - Run the **Command Prompt (CMD)** with **Administrator privileges**:



2 - Go to the Apache `bin` folder to install the **httpd.exe** executable file:

```
cd C:\Apache24\bin\httpd.exe
```

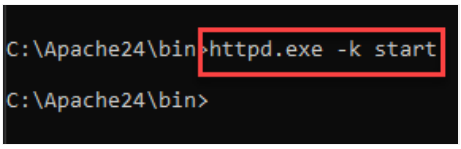
3 - Install Apache with the command below:

`httpd.exe -k install`

Wait for the message **"The 'Apache2.4' service is successfully installed."** to proceed.

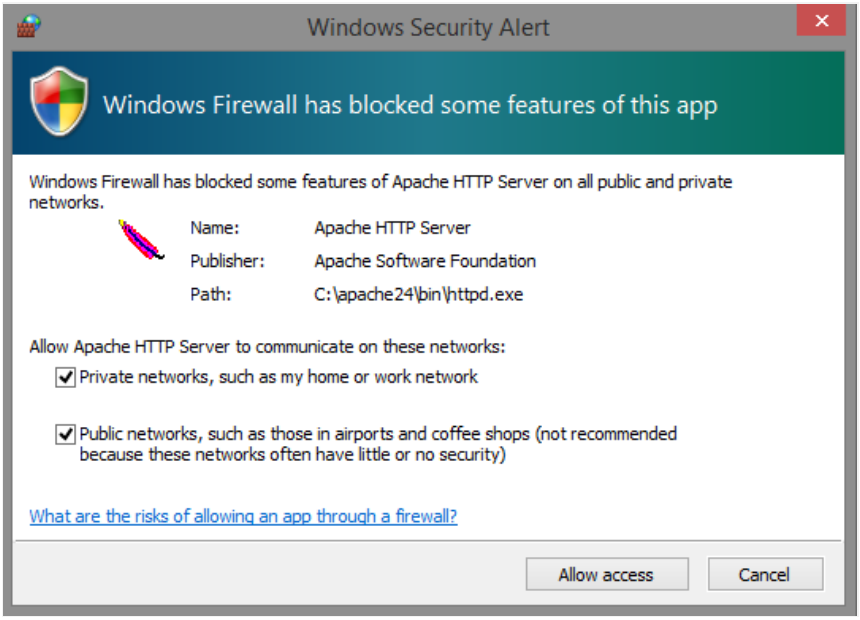
4 - Start the Apache service and verify if it is working.

Use the `httpd.exe -k start` to start your Apache service.



5 - Grant execution permission to Apache on your Firewall:

Select the two options available to grant permission, so Apache can function properly.



6 - Check if Apache is active.

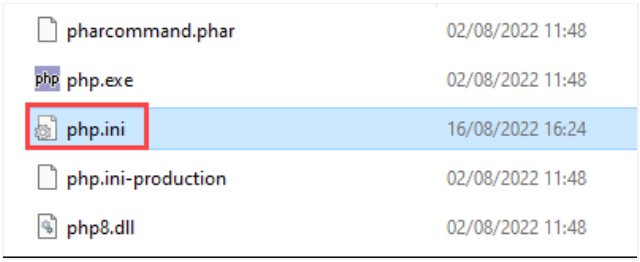
To verify that the service is active, simply visit the Apache test page in your browser. You can do it in two ways:

`127.0.0.1` or `localhost`

Configure the PHP 8.1

With Apache active and functional, you now need to define and configure the **PHP.ini** file so those database extensions and libraries are available for use by Scriptcase.

1 - Access the PHP folder in `C:\` and rename the **php.ini-development** file to **php.ini**.

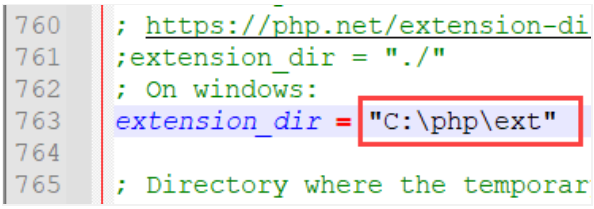


2 - Access the **php.ini** file with a text editor using Administrative privileges and search for the `;extension_dir = "ext"` directive.

Assign value **"C:\php\ext"** to this directive and remove **;**.

Example:

`extension_dir = "C:\php\ext"`



3 - Enable the extensions listed below, by uncommenting them **(Remove the semicolon `;` at beginning of line)**.

Default Extensions

- `extension=bz2`
- `extension=curl`
- `extension=gd2`
- `extension=gettext`
- `extension=imap`
- `extension=ldap`
- `extension=fileinfo`
- `extension=mbstring`
- `extension=openssl`
- `extension=exif`
- `extension=xsl`
- `extension=soap`

Database Extensions:

Extensions	Driver enabled in Scriptcase
`extension=mysqli`	MySQLi
`extension=odbc`	DB2 ODBC GENERIC, DB2 ODBC GENERIC 6, Generic ODBC, MS Access ODBC, MS SQL Server ODBC, Oracle ODBC, Progress
`extension=pdo_mysql`	MySQL PDO
`extension=pdo_odbc`	DB2 PDO ODBC, Progress PDO ODBC, Sybase PDO ODBC
`extension=pdo_pgsql`	PostgreSQL PDO
`extension=pdo_sqlite`	SQLite PDO
`extension=pgsql`	PostgreSQL 6.3 or below, PostgreSQL 6.4 or above, PostgreSQL 7 ou Acima

NOTE: Some database extensions will depend on third party drivers to be enabled. Click here to access configuration instructions for your desired database.

4 - Set the recommended minimum value of these PHP directives listed below for Scriptcase to work proprely. Search for the directives and assign the value according to this example:

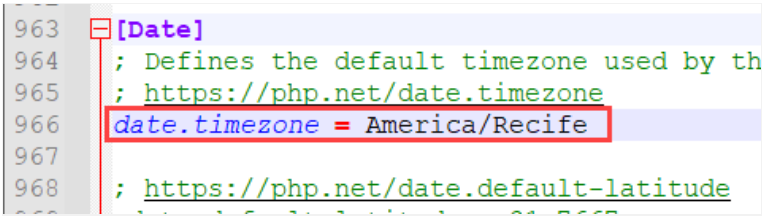
- `max_execution_time = 3600`
- `max_input_time = 3600`
- `max_input_vars = 10000`
- `memory_limit = 1024M`
- `post_max_size = 1024M`
- `upload_max_filesize = 1024M`
- `max_file_uploads = 200`
- `short_open_tag = On`

IMPORTANT: By default PHP can disable some functions that are important for Scriptcase to function properly. Search for the **disable_functions** directive and make sure it is as follows: `disable_functions=` . Click here to access the list of functions required for Scriptcase to function properly.

5 - Set up PHP **TimeZone** according to your region. You must use the value available in PHP documentation. Search for the **date.timezone** line, uncomment it and edit it according to the selected TimeZone:

Example:

`date.timezone = America/Recife`



6 - Set the folder where temporary files will be stored. Search for the `;session.save_path` line, uncomment it, and enter the path to your temporary folder.

Example:

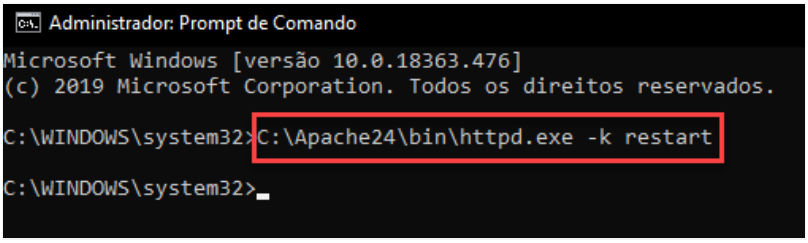
`session.save_path = "C:\Windows\Temp"`

```
;     session.save_path = "N;MODE;/path"
;
; where MODE is the octal representation of
; does not overwrite the process's umask.
; https://php.net/session.save-path
session.save_path = "C:\Windows\Temp"

; Whether to use strict session mode.
; Strict session mode does not accept an un
```

7 - Save all changes made to the `php.ini` file.

8 - Restart the Apache service using the Command Prompt **(CMD)** with Administrator privileges the following command:

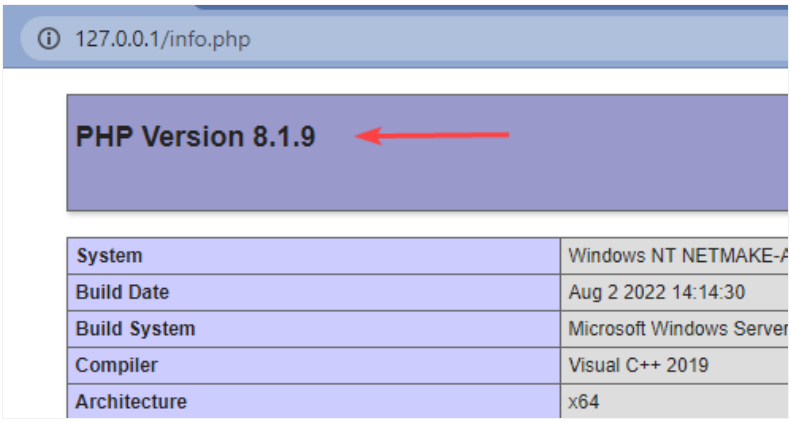
`C:\Apache24\bin\httpd.exe -k restart`



9 - Verify changes made through the **info.php** file. You need to create this file and place it in the `C:\Apache24\htdocs` directory with the following content:

```
<?php
   phpinfo();
?>
```

After this, check the generated page in your browser by going to the URL `127.0.0.1/info.php` .



NOTE: For more information about the **phpinfo()** function, check the documentation do PHP.

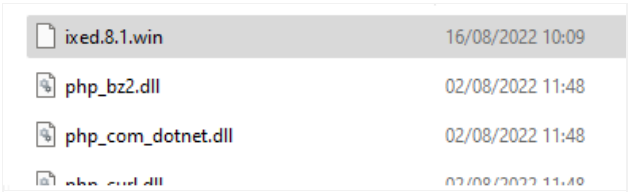
Enabling the SourceGuardian Loader

Before you start the Scriptcase installation, you need to enable the loader responsible for encrypting Scriptcase in PHP. Follow the steps below to perform this action successfully.

1 - Extract the **SourceGuardian Loader** file downloaded at the beginning of this documentation.

If you haven't downloaded it yet click here and download the file according to the architecture used.

2 - Copy the **ixed.8.1.win** file and paste it into the PHP extensions directory `C:\php\ext` .



3 - Edit the **php.ini** file and below the last line enter the path to the extension within the `zend_extension` parameter:

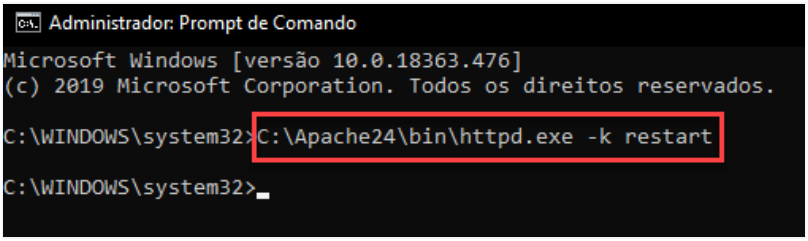
Example:

```
[SourceGuardian] zend_extension=C:\php\ext\ixed.8.1.win
```

```
;ffi.enable=preload

; List of headers files to preload, wildcard patterns allowed.
;ffi.preload=

[SourceGuardian] zend_extension=C:\php\ext\ixed.8.1.win
```

4 - Restart the Apache service using the Command Prompt **(CMD)** with Administrator privileges the following command:

```
C:\Apache24\bin\httpd.exe -k restart
```



Manual Scriptcase Installation

Listed below are the steps needed to do a Scriptcase manual installation. **To do the steps, it is needed that you have a web server configured in your machine**.

1 - Download the Scriptcase **(.zip)** directly from the download page from our website.

2 - Extract the **.zip** file and rename the extracted folder to `scriptcase` .

3 - Move the **scriptcase** to your server root. **Depending of your operating system and the installation the path can be different from the example.** However, by default the paths are:

Windows	Server
`C:\Apache24\htdocs\`	Apache
`C:\inetpub\wwwroot\`	IIS

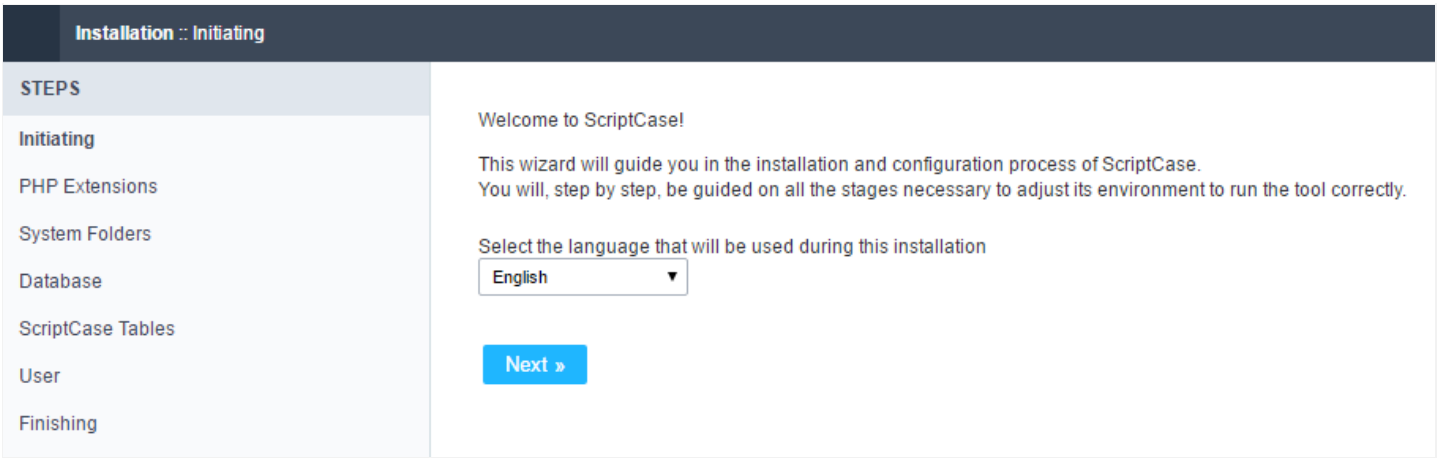
Linux	Server
/var/www/html/	Linux Local
/home/$(whoami)/public_html/	Linux Server

macOS	Server
/Library/WebServer/Documents	Apache

4 - Access the Scriptcase using your browser:

`127.0.0.1/scriptcase`

- **By default the selected language is English.**



5 - Check the extensions required for Scriptcase to function and database connection modules that are enabled.

Check if the necessary PHP extensions are loaded.

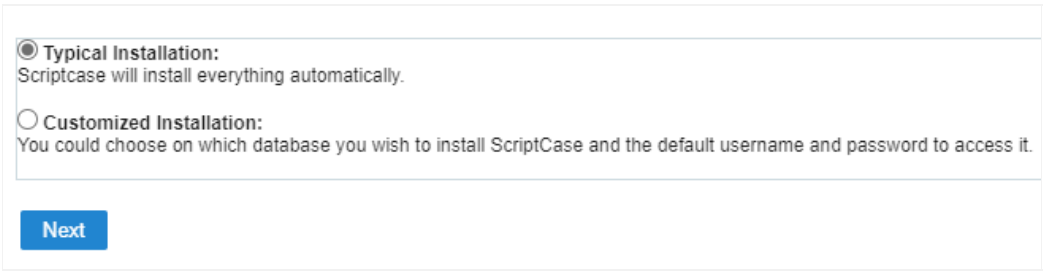
Required Extensions.		Action.
✓	GD	This extension is used to create charts and manipulate images
✓	MBSTRING	This extension is used to convert special chars
✓	SimpleXML	The SimpleXML extension provides a very simple and easily usable toolset to convert XML to an object that can be processed with normal property selectors and array iterators.
✓	ZIP	This extension is used to create and extract files from a ZIP archive
✓	ZLIB	This extension is used to extract files from a ZIP archive
✓	JSON	This extension implements the data interchange format JavaScript Object Notation (JSON).

Database Extensions.			Action.
✓	com_dotnet	Generic ADO, MS Access ADO, MS SQL Server ADO	
✗	ibm_db2	DB2, DB2 ODBC NATIVE	Enable.
✓	interbase	Firebird, Interbase 6, Interbase 6.5	
✗	mssql	MS SQL Server 7	Enable.
✗	mysql	MaxSQL, MySQL (Non-Transactional), MySQL (Transactional)	Enable.
✓	mysqli	MySQLi	
✓	oci8	Oracle 8, Oracle 8 Portable, Oracle 8.0.5	
✓	odbc	DB2 ODBC GENERIC, DB2 ODBC GENERIC 6, Generic ODBC, MS Access ODBC, MS SQL Server ODBC, Oracle ODBC, Progress	
✗	oracle	Oracle 7	Enable.
✓	pdo_dblib	Sybase PDO DBLIB	
✓	pdo_firebird	Firebird PDO	
✗	pdo_ibm	PDO IBM	Enable.
✗	pdo_informix	Informix PDO	Enable.

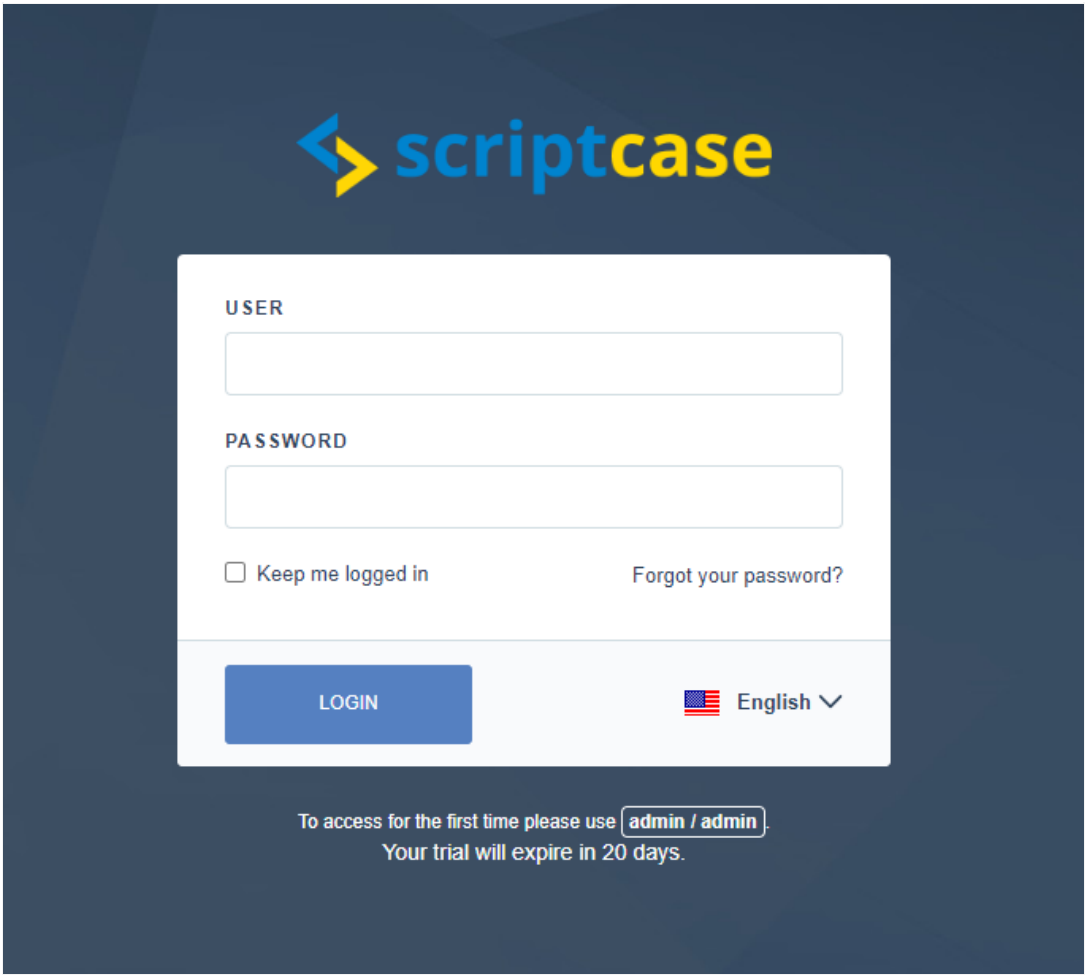
Typical Installation

Check the steps below to complete the manual installation of Scriptcase in a typical way.

NOTE: By selecting this installation, Scriptcase will set your environment by default. Defining the installation database and the login user in Scriptcase.



After clicking proceed, you will be redirected to the Scriptcase Login page.



User

In this field, it is necessary to inform the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me connected

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

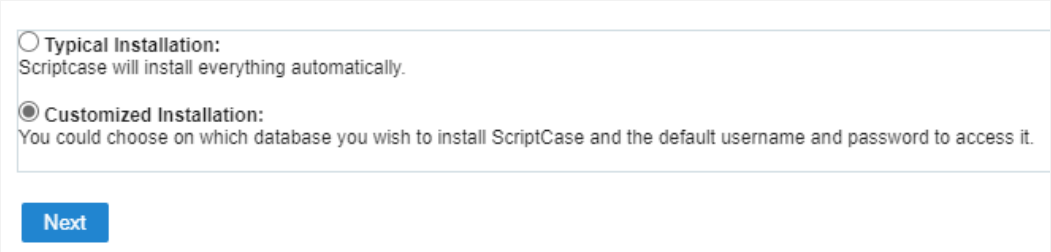
By clicking this option, it is possible to recover the password of the informed user. An email will be sent to the user in question so they can access the recovery steps.

Important: You can only perform password recovery if you have configured your SMTP information. To access our documentation showing how to perform this configuration click here.

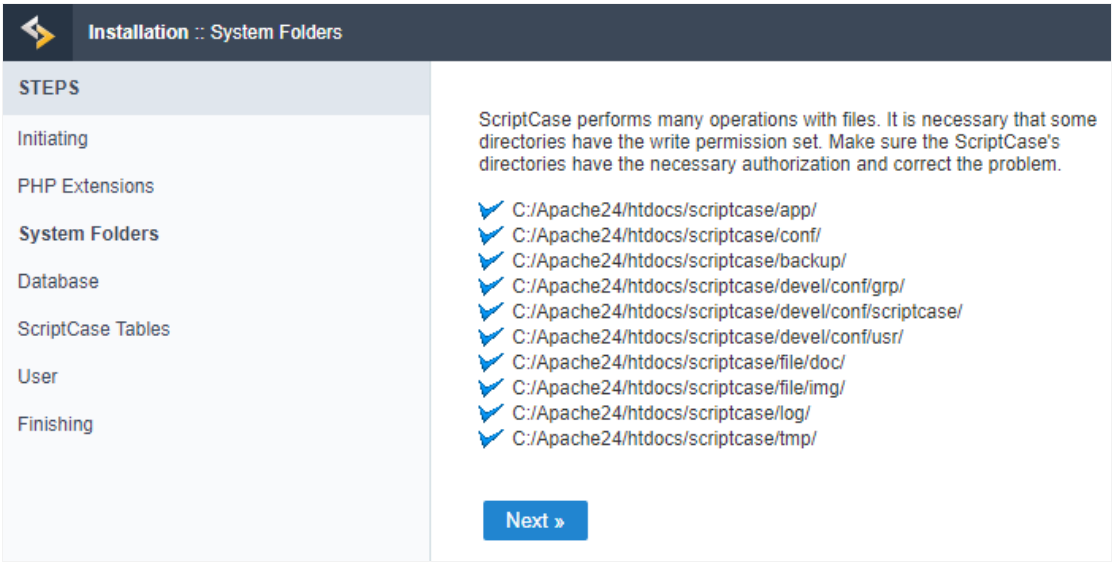
Custom Installation

Check the steps below to complete the manual installation of Scriptcase in a custom manner.

NOTE: By selecting this installation, you will be responsible for setting up your environment. Defining the installation database and the login user for Scriptcase.

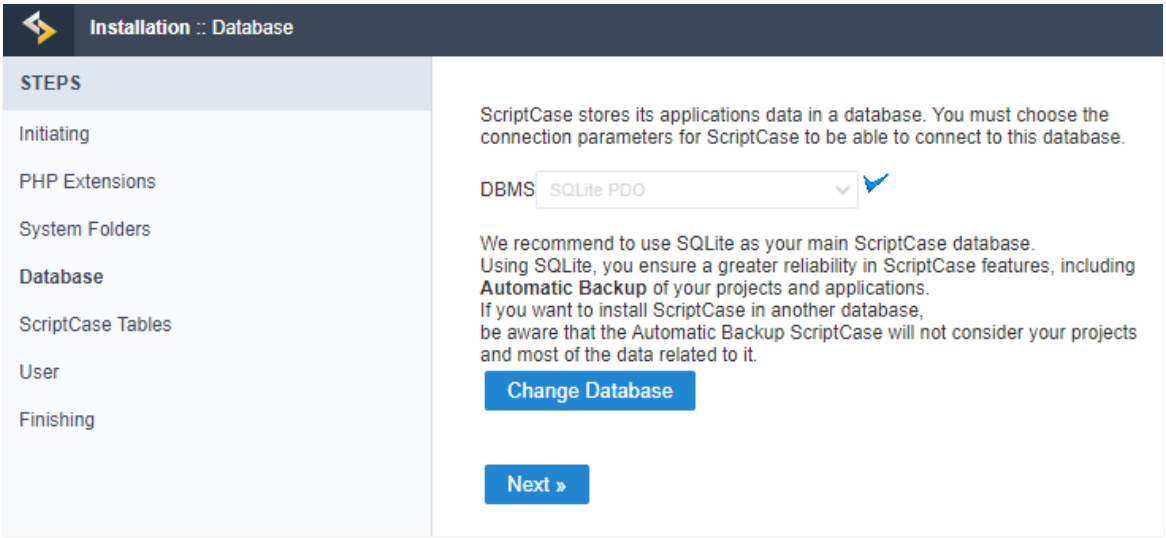


1- Verify that the directories required by Scriptcase have the correct permissions. If any directory listed does not have the active check icon , correct permission for this directory.

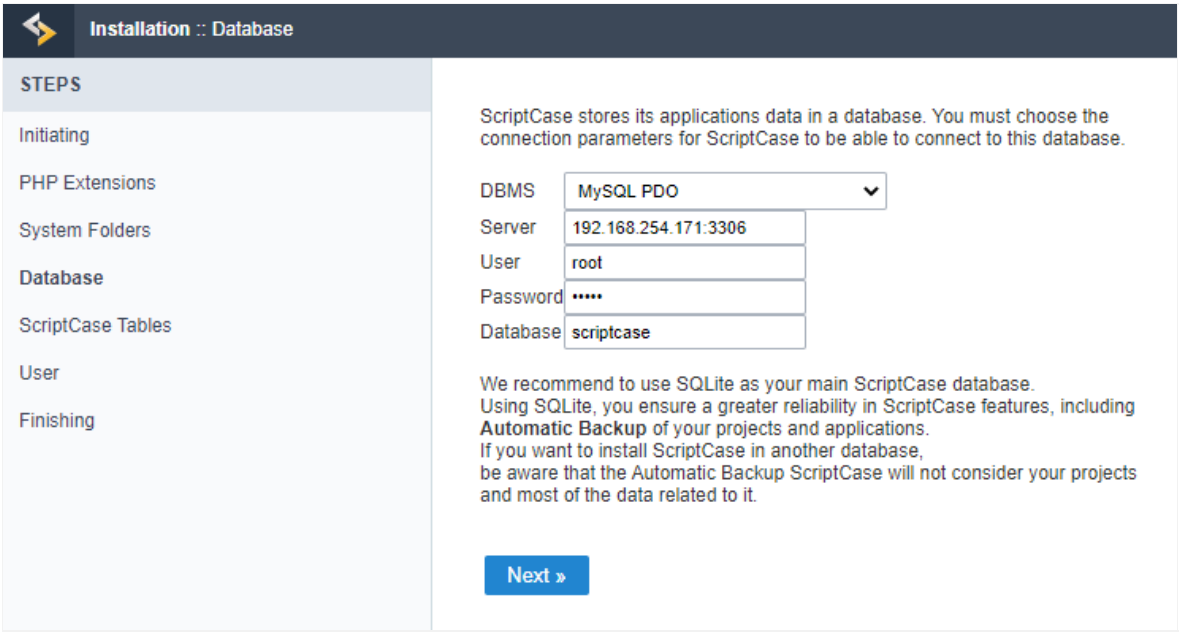


Click proceed to proceed with the installation process .

2- Choose the database where the Scriptcase database will be installed. By default, Scriptcase recommends that it be installed with the **SQLite** database.



- You can change the default database by clicking the **Change Database** button. By clicking this option you will need to set the following options:



DBMS

In this option, you will select the Driver to connect to your database.

Server

In this option, you will inform the server to connect to the desired database.

User

You need to inform a user who has access to the database you want Scriptcase to be installed on.

Password

You must enter the authentication password corresponding to the entered user.

Database

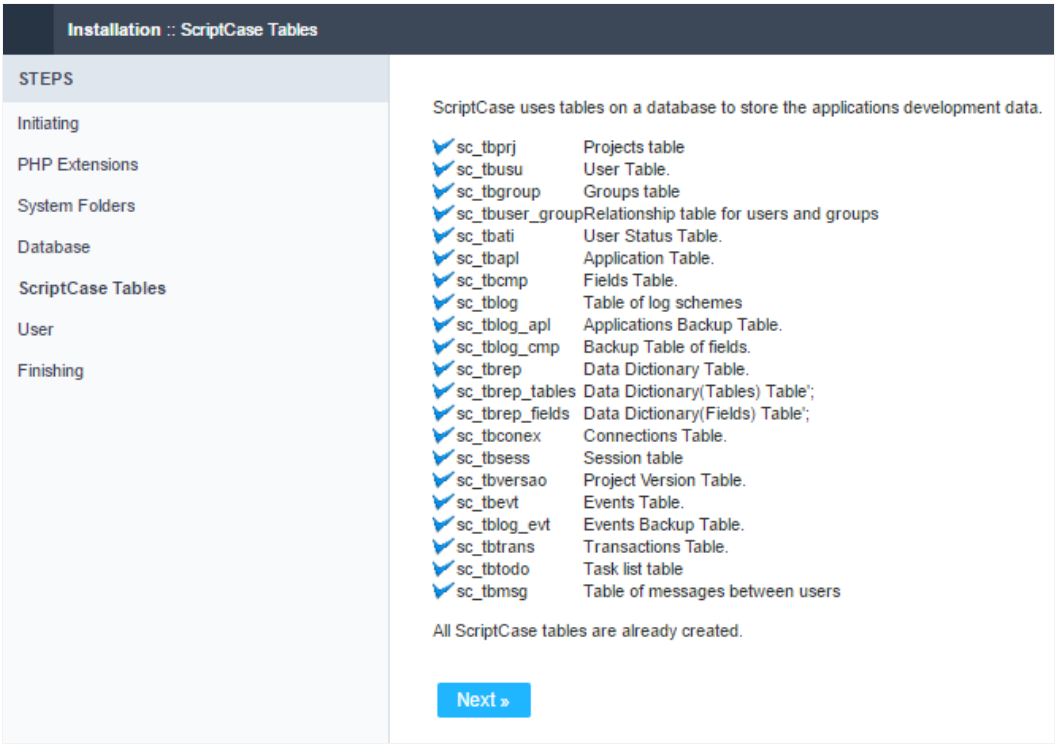
You must enter the name of the database where Scriptcase will be installed.

Example:

`scriptcase`

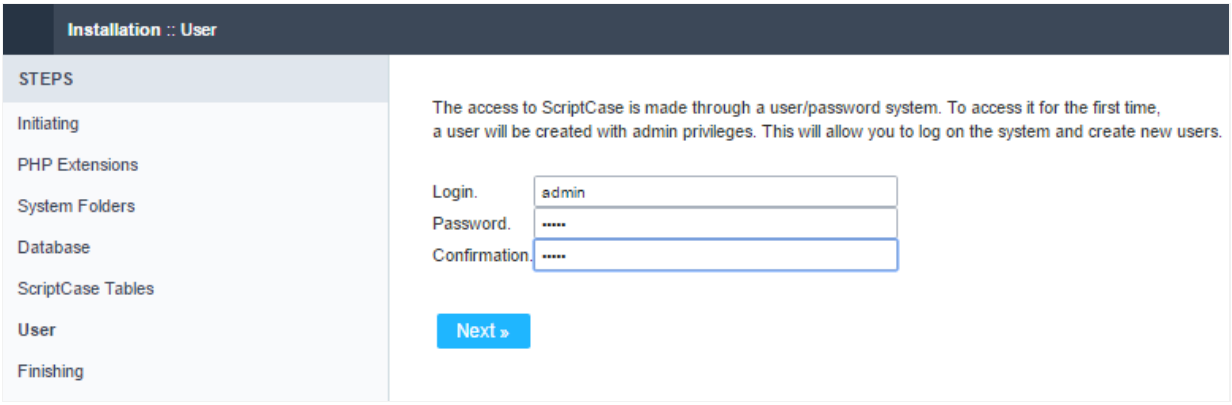
Clique em prosseguir para avançar no processo de instalação.

3 - Verifique se a criação das tabelas da base de dados do Scriptcase ocorreu com sucesso. As tabelas serão marcadas com o ícone de sucesso na lateral.



Click proceed to proceed with the installation process.

_4 -__ Set the default user who will have access to Scriptcase.



Login

In this option, you will enter the default Scriptcase User Login.

Password

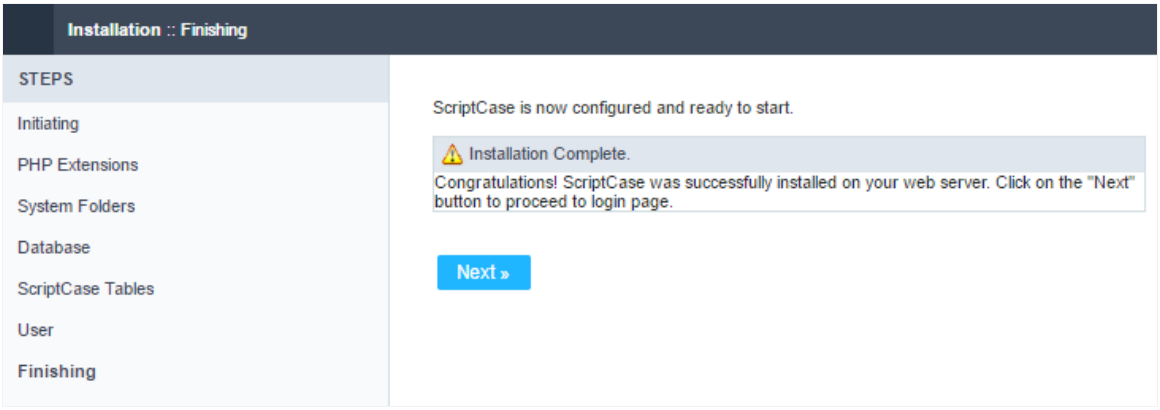
In this option, you must enter the password that will correspond to the user informed.

Confirmation

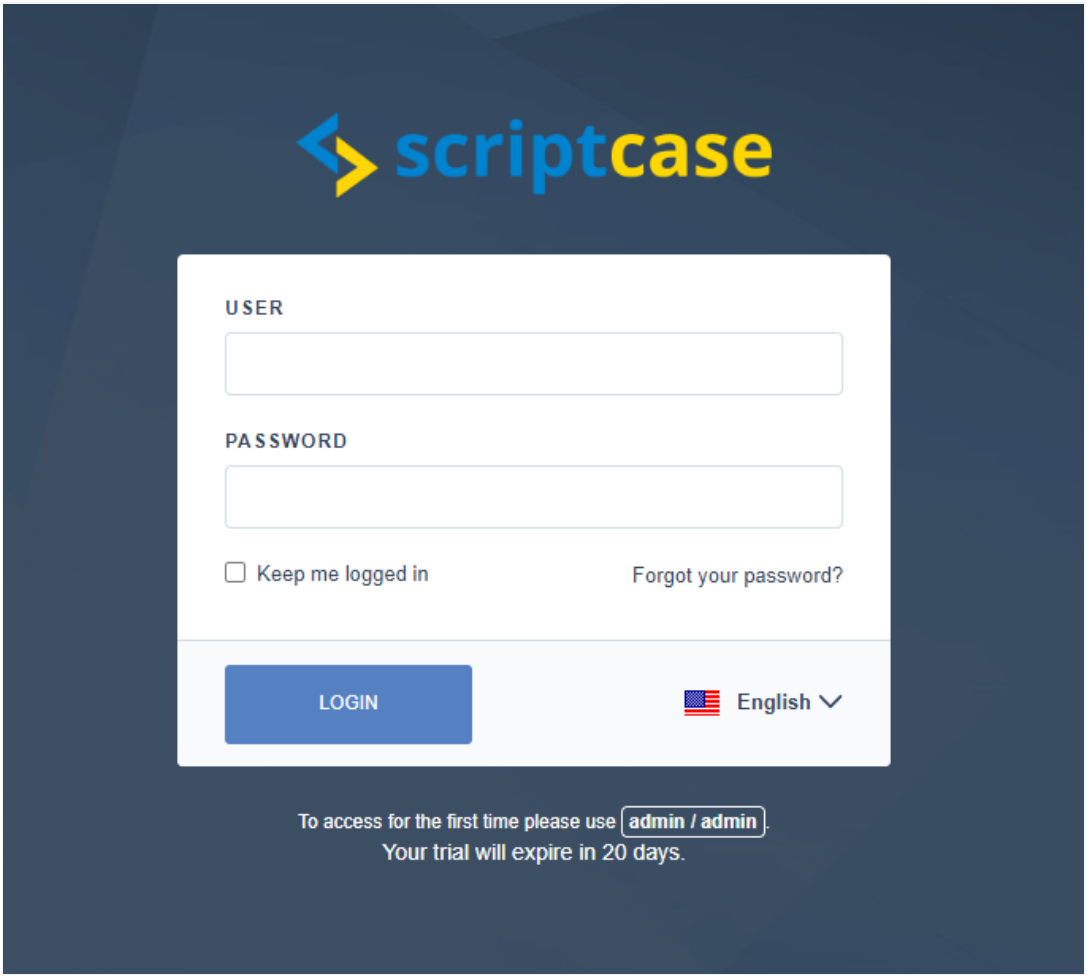
You must repeat the previously entered password for successful user creation.

Click proceed to proceed with the installation process.

5 - Complete Scriptcase custom installation on your web server.



After clicking proceed, you will be redirected to the Scriptcase Login page.



User

In this field, it is necessary to inform the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me connected

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

By clicking this option, it is possible to recover the password of the informed user. An email will be sent to the user in question so they can access the recovery steps.

Important: You can only perform password recovery if you have configured your SMTP information. To access our documentation showing how to perform this configuration click here.

Installing PHP 8.1 - Linux

This documentation will teach all the steps necessary for the installation of the PHP 8.1, Apache 2.4 and the Scriptcase using the manual installation in the Linux environment. **Doing this kind of installation you will be responsible for configuring all your environment settings and the database extensions as well.**

Listed below are the operational systems approved for the Scriptcase usage. **Only these distributions below and their derivatives are approved for the Scriptcase installation.**

Check the list of the supported operating systems:

- RedHat
- Debian\Ubuntu
- Suse\OpenSuse

NOTE: In case you need to make a manual installation of the Scriptcase using **PHP 7.3** in Linux, access the documentation in our knowledge base.

Prerequisites

WARNING: If you already have a Web Server with Apache and PHP 8.1 configured, just go to the SourceGuardian configuration and proceed to manual installation.

To proceed with the installation, you will need to download some files.

Necessaries Files:

- SourceGuardian for **Linux (x86 ou x64)**: Click Here
- Scriptcase **(.zip)**: Click Here

NOTE: Before continue with the installation is recommended that you do not have any other web environment (as XAMP, Zend Server, etc) installed in your operational system. **It is recommended to know Linux/Unix environment to use the terminal.**

Configuring the PHP - Ubuntu 22.04\Debian 10

Listed below are the necessaries commands for the manual installation of the PHP in the Ubuntu 22.04 or Debian 10. Check correctly the name of the packages in your terminal.

1 - Before run any other command, update your operating system with the following commands:

```
sudo apt-get update
sudo apt-get upgrade
```

2 - Run the command `sudo apt-get install php8.1` to install the PHP 8.1. After that, accept the dependencies installation.

If you are using any Ubuntu version before 19.10, it's necessary to add a repository in your operating system to access the PHP 8.1 packages. See how you can do it below:

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

If you are using any Debian version before Debian 10, it's necessary to add a repository in your operating system to access the PHP 8.1 packages. See how you can do it below:

```
sudo apt -y install lsb-release apt-transport-https ca-certificates
```

```
sudo wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
echo "deb https://packages.sury.org/php/ $(lsb_release -sc) main" | sudo tee /etc/apt/sources.list.d/php8.1.list
```

`sudo apt-get update`

`sudo apt-get install php8.1`

Example:

```
netmake@netmake-Vostro-3560:~$ sudo apt-get install php8.1
Lendo listas de pacotes... Pronto
Construindo árvore de dependências
Lendo informação de estado... Pronto
Os pacotes adicionais seguintes serão instalados:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php8.1 libapr1 libaprutil1 libaprutil1-dbd-sqlite
  libaprutil1-ldap liblua5.2-0 php-common php8.1-cli php8.1-common php8.1-opcache php8.1-readline
Pacotes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom php-pear
Os NOVOS pacotes a seguir serão instalados:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php8.1 libapr1 libaprutil1 libaprutil1-dbd-sqlite
  libaprutil1-ldap liblua5.2-0 php-common php8.1 php8.1-cli php8.1-common php8.1-opcache php8.1-readline
0 pacotes atualizados, 16 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
É preciso baixar 0 B/6.586 kB de arquivos.
Depois desta operação, 29,3 MB adicionais de espaço em disco serão usados.
Você quer continuar? [S/n] S
```

3 - Check the PHP version running the command: `php -v`

Example:

```
netmake@netmake-Vostro-3560:~$ php -v
PHP 8.1.8 (cli) (built: Jul 11 2022 08:29:57) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.8, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.8, Copyright (c), by Zend Technologies
```

4 - Now, install all the extensions below that are required by Scriptcase:

Default Extensions:

- `sudo apt-get install php8.1-curl`
- `sudo apt-get install php8.1-gd`
- `sudo apt-get install php8.1-bcmath`
- `sudo apt-get install php8.1-cgi`
- `sudo apt-get install php8.1-ldap`
- `sudo apt-get install php8.1-mbstring`
- `sudo apt-get install php8.1-xml`
- `sudo apt-get install php8.1-soap`
- `sudo apt-get install php8.1-xsl`
- `sudo apt-get install php8.1-zip`

Database Extensions:

Extension	Driver enabled in Scriptcase
`sudo apt-get install php8.1-pgsql`	PostgreSQL PDO, PostgreSQL 6.3 or below, PostgreSQL 6.4 or above, PostgreSQL 7 or above
`sudo apt-get install php8.1-mysql`	MySQLi, MySQL PDO
`sudo apt-get install php8.1-sqlite3`	SQLite PDO
`sudo apt-get install php8.1-interbase`	Firebird, Interbase 6, Interbase 6.5, Firebird PDO
`sudo apt-get install php8.1-odbc`	DB2 ODBC GENERIC, DB2 ODBC GENERIC 6, Generic ODBC, MS Access ODBC, Oracle ODBC, Progress, DB2 PDO ODBC, Progress PDO ODBC, Sybase PDO ODBC
`sudo apt-get install php8.1-sybase`	DBLIB, Sybase PDO DBLIB

NOTE: Some database extensions will depend on third party drivers to be enabled. Click here to access the setup instructions for your database.

5 - Access the **php.ini** file in the directory `/etc/php/8.1/apache2/` and set the recommended values for the PHP directives listed below required by the Scriptcase.

Search for theses directives and set the values as the example:

- `max_execution_time = 3600`
- `max_input_time = 3600`
- `max_input_vars = 10000`
- `memory_limit = 1024M`
- `post_max_size = 1024M`
- `upload_max_filesize = 1024M`
- `max_file_uploads = 200`
- `short_open_tag = On`

6 - By default, the PHP disable some functions that are required by the Scriptcase. Click here to access the list of these functions.

- Change the line **disable_functions** as the example below:

`disable_functions =`

Example:

```
; This directive allows you to disable certain functions for security reasons.
; It receives a comma-delimited list of function names.
; http://php.net/disable-functions
disable_functions =
```

7 - Set up the PHP **TimeZone** according your region. Must use the values available in the PHP documentation. Search for the line **date.timezone** in your **php.ini**, uncomment and edit according with the TimeZone selected.

Example:

`date.timezone = America/Recife`

8 - Set up the folders that will store the temporary files. Search for the line `;session.save_path` , uncomment and set the path to your temporary folder.

Example:

```
session.save_path = "/tmp"
```

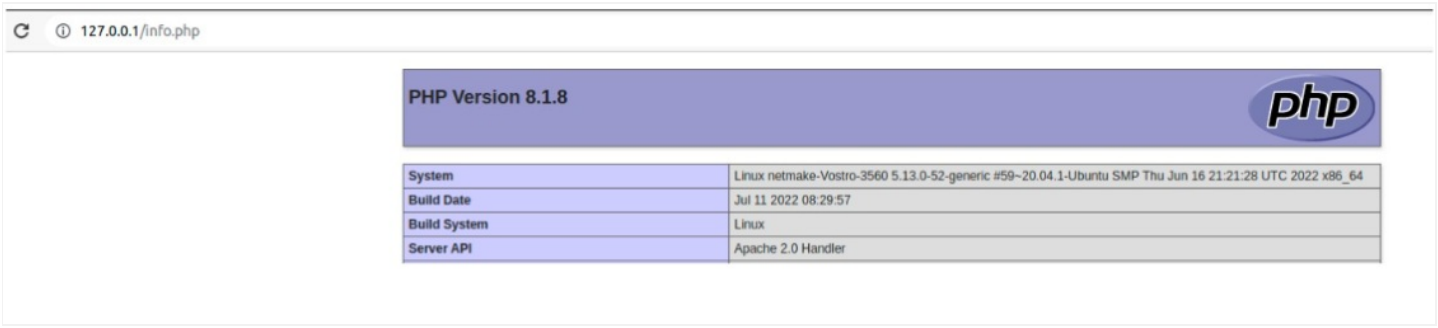
9 - Save all the changes made and restart the Apache service using the following command:

```
sudo service apache2 restart
```

10 - Check if the changes were made using the file **info.php**. You will need to create this file and place it in the `/var/www/html/` path with the following content.

```
<?php
phpinfo();
?>
```

After that, check using your browser the page generated accessing the URL `127.0.0.1/info.php` .



NOTE: For more information about the **phpinfo()**, check the PHP documentation.

Enabling the SourceGuardian

Before starting the Scriptcase installation, you need to enable the loader used for the Scriptcase encryption in your PHP. Follow the steps below showing how to dot it.

1 - Access the Downloads folder and extract the file downloaded referent to the **SourceGuardian** at the beggining of this documentation.

Example:

```
cd /home/User/Downloads
```

```
sudo chmod 777 loaders.linux-x86_64.zip && unzip loaders.linux-x86_64.zip
```

2 - Copy the file **ixed.8.1.lin** and past in the **extension_dir** directory of your PHP. We will use the path `/usr/lib/php/20210902` for this example.

Example:

Ubuntu 22.04\Debian 10
`/usr/lib/php/20180731`

```
sudo cp ixed.8.1.lin /usr/lib/php/20210902
```

3 - Edit the **php.ini** file and below the last line set the path to the extension in the `zend_extension` parameter:

Example:

```
zend_extension = "/usr/lib/php/20210902/ixed.8.1.lin"
```

Ubuntu 22.04\Debian 10
`/etc/php/8.1/apache2/`

`sudo nano /etc/php/8.1/apache2/php.ini`

`zend_extension = "/usr/lib/php/20210902/ixed.8.1.lin"`

4 - Restart the Apache service using the following command:

Ubuntu 22.04\Debian 10	
`sudo service apache2 restart`	`sudo systemctl restart httpd`

Scriptcase Manual Installation

Listed below are the steps needed to do a Scriptcase manual installation. **To do the steps, it is needed that you have a web server configured in your machine**.

1 - Download the Scriptcase **(.zip)** directly from the download page from our website.

2 - Extract the **.zip** file and rename the extracted folder to `scriptcase` .

3 - Move the **scriptcase** to your server root. **Depending of your operating system and the installation the path can be different from the example.** However, by default the paths are:

Windows	Server
`C:\Apache24\htdocs\`	Apache
`C:\inetpub\wwwroot\`	IIS

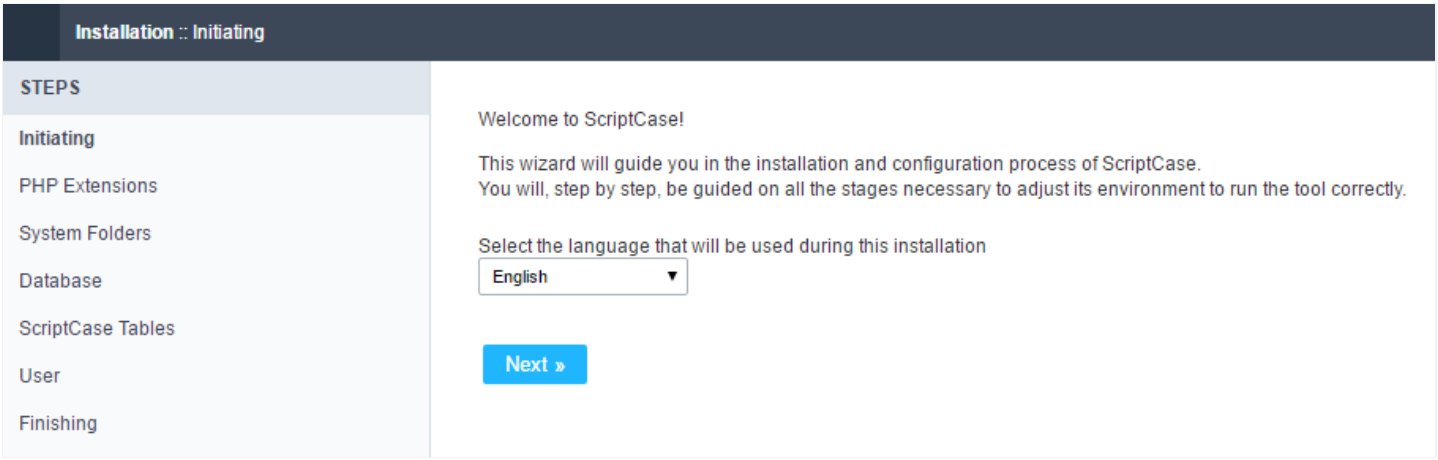
Linux	Server
`/var/www/html/`	Linux Local
`/home/$(whoami)/public_html/`	Linux Server

macOS	Server
`/Library/WebServer/Documents`	Apache

4 - Access the Scriptcase using your browser:

127.0.0.1/scriptcase

- **By default the selected language is English.**



5 - Check the extensions required for Scriptcase to function and database connection modules that are enabled.

Check if the necessary PHP extensions are loaded.

Required Extensions.		Action.
GD	This extension is used to create charts and manipulate images	
MBSTRING	This extension is used to convert special chars	
SimpleXML	The SimpleXML extension provides a very simple and easily usable toolset to convert XML to an object that can be processed with normal property selectors and array iterators.	
ZIP	This extension is used to create and extract files from a ZIP archive	
ZLIB	This extension is used to extract files from a ZIP archive	
JSON	This extension implements the data interchange format JavaScript Object Notation (JSON).	

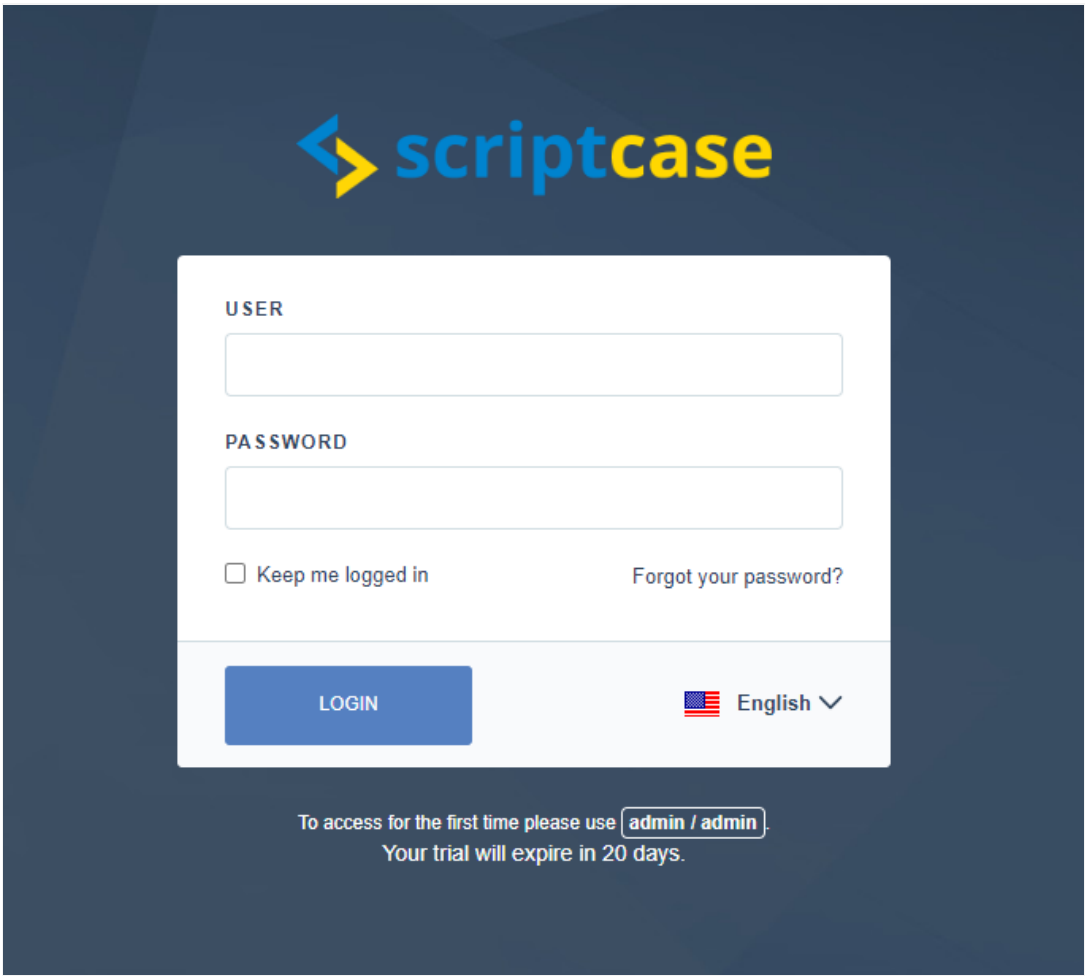
Database Extensions.		Action.
com_dotnet	Generic ADO, MS Access ADO, MS SQL Server ADO	
ibm_db2	DB2, DB2 ODBC NATIVE	Enable.
interbase	Firebird, Interbase 6, Interbase 6.5	
mssql	MS SQL Server 7	Enable.
mysql	MaxSQL, MySQL (Non-Transactional), MySQL (Transactional)	Enable.
mysqli	MySQLi	
oci8	Oracle 8, Oracle 8 Portable, Oracle 8.0.5	
odbc	DB2 ODBC GENERIC, DB2 ODBC GENERIC 6, Generic ODBC, MS Access ODBC, MS SQL Server ODBC, Oracle ODBC, Progress	
oracle	Oracle 7	Enable.
pdo_dblib	Sybase PDO DBLIB	
pdo_firebird	Firebird PDO	
pdo_ibm	PDO IBM	Enable.
pdo_informix	Informix PDO	Enable.

Typical Installation

Check the steps below to complete the manual installation of Scriptcase in a typical way.

NOTE: By selecting this installation, Scriptcase will set your environment by default. Defining the installation database and the login user in Scriptcase.

After clicking proceed, you will be redirected to the Scriptcase Login page.



User

In this field, it is necessary to inform the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me connected

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

By clicking this option, it is possible to recover the password of the informed user. An email will be sent to the user in question so they can access the recovery steps.

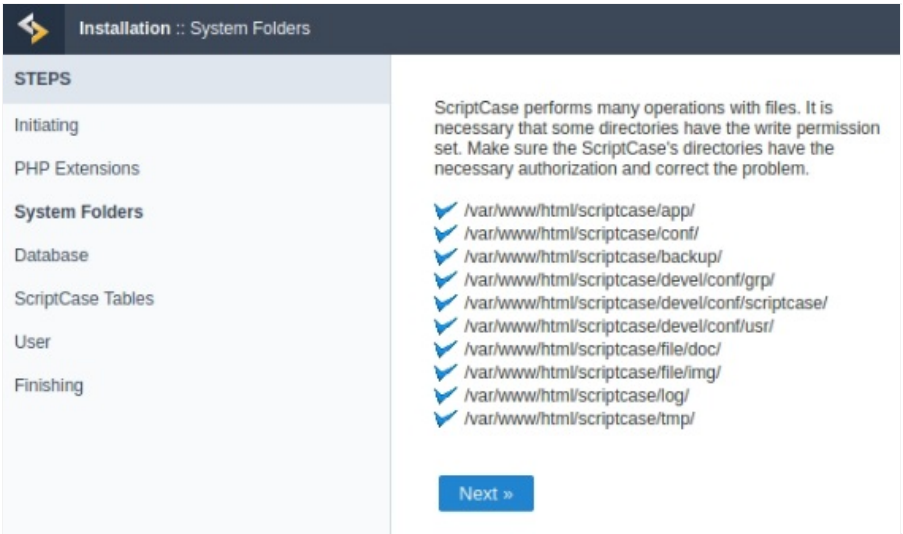
Important: You can only perform password recovery if you have configured your SMTP information. To access our documentation showing how to perform this configuration click here.

Customized Installation

Check the steps below to complete the manual installation of Scriptcase in a custom manner.

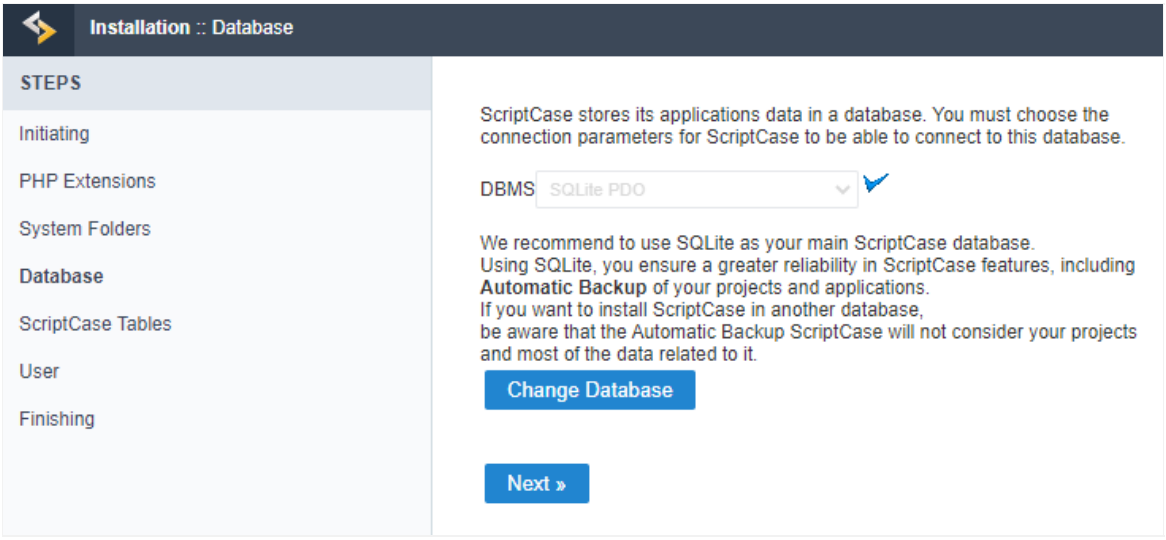
NOTE: By selecting this installation, you will be responsible for setting up your environment. Defining the installation database and the login user for Scriptcase.

1- Verify that the directories required by Scriptcase have the correct permissions. If any directory listed does not have the active check icon , correct permission for this directory.

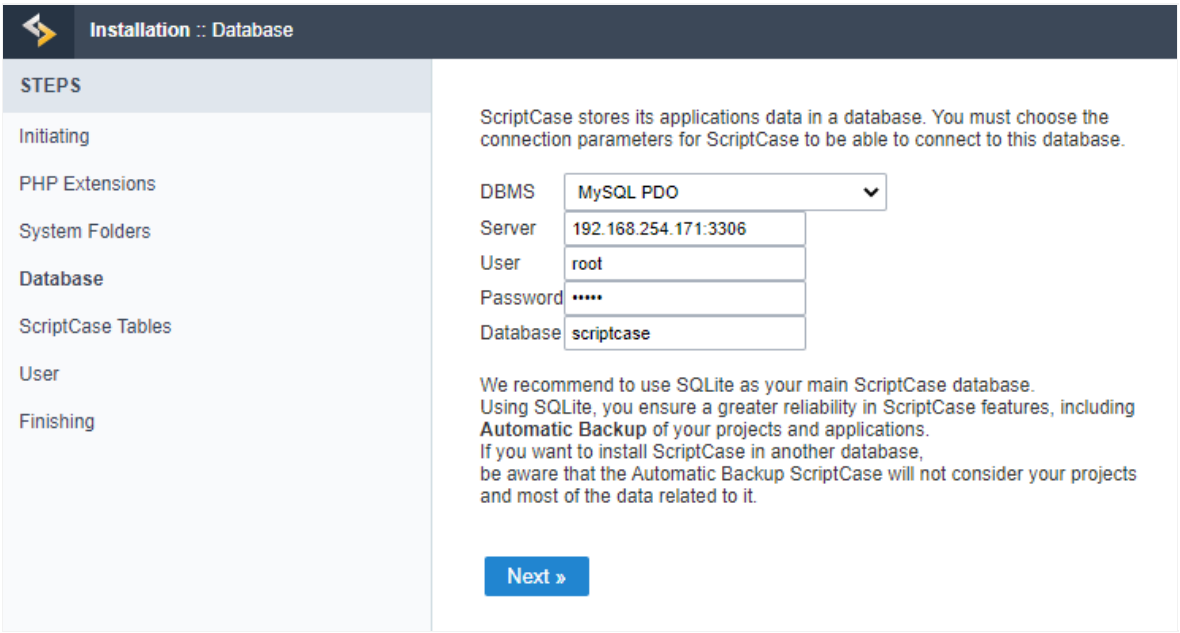


Click proceed to proceed with the installation process .

2- Choose the database where the Scriptcase database will be installed. By default, Scriptcase recommends that it be installed with the **SQLite** database.



- You can change the default database by clicking the **Change Database** button. By clicking this option you will need to set the following options:



DBMS

In this option, you will select the Driver to connect to your database.

Server

In this option, you will inform the server to connect to the desired database.

User

You need to inform a user who has access to the database you want Scriptcase to be installed on.

Password

You must enter the authentication password corresponding to the entered user.

Database

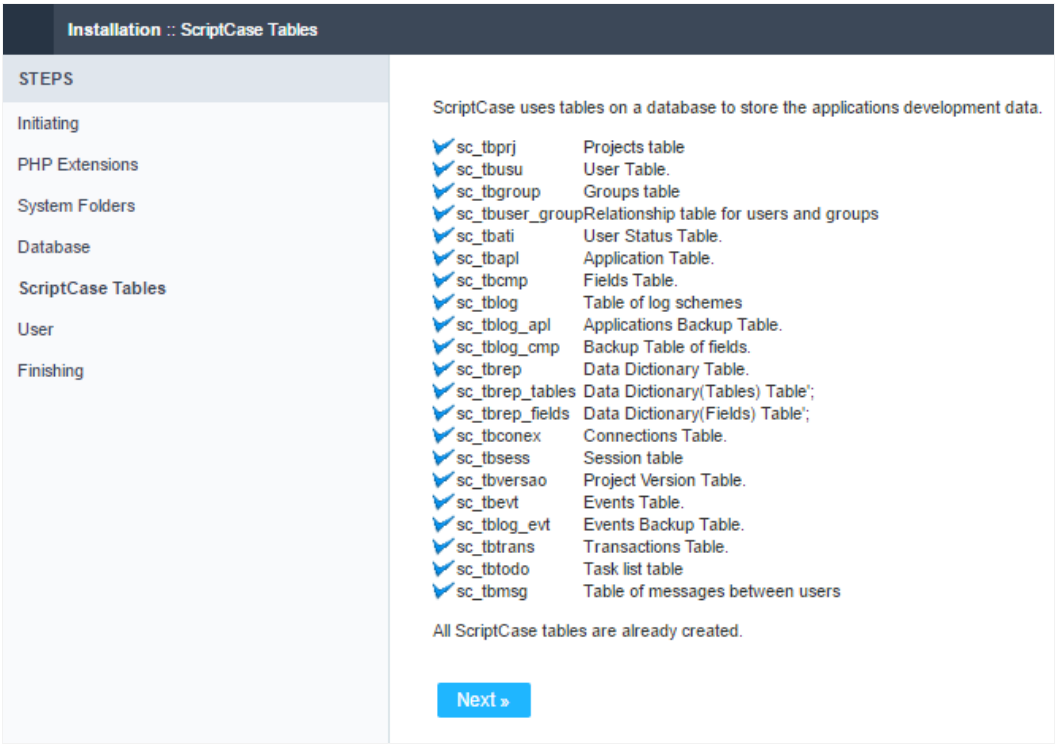
You must enter the name of the database where Scriptcase will be installed.

Example:

`scriptcase`

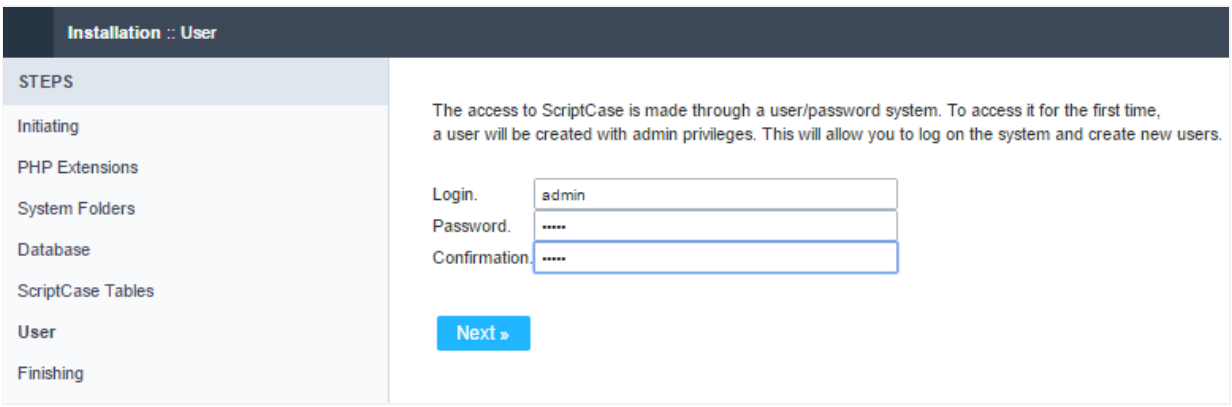
Clique em prosseguir para avançar no processo de instalação.

3 - Verifique se a criação das tabelas da base de dados do Scriptcase ocorreu com sucesso. As tabelas serão marcadas com o ícone de sucesso na lateral.



Click proceed to proceed with the installation process.

_4 -__ Set the default user who will have access to Scriptcase.



Login

In this option, you will enter the default Scriptcase User Login.

Password

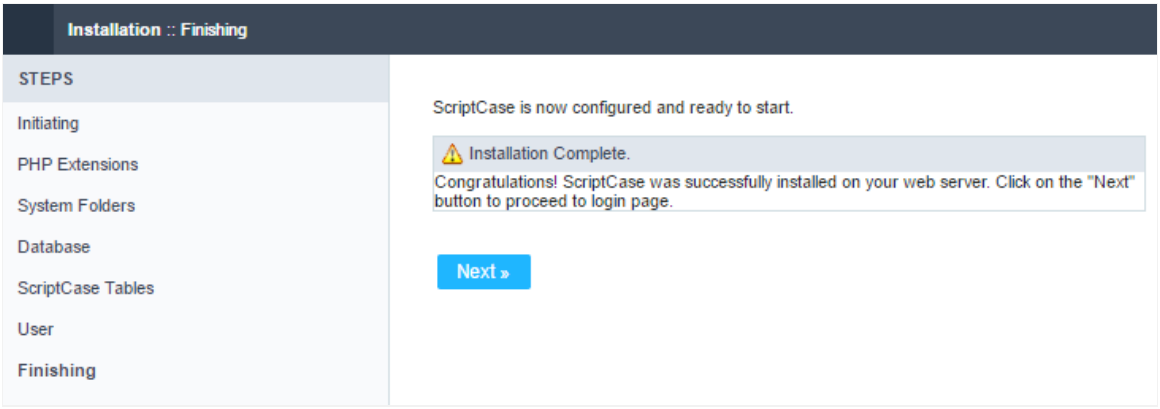
In this option, you must enter the password that will correspond to the user informed.

Confirmation

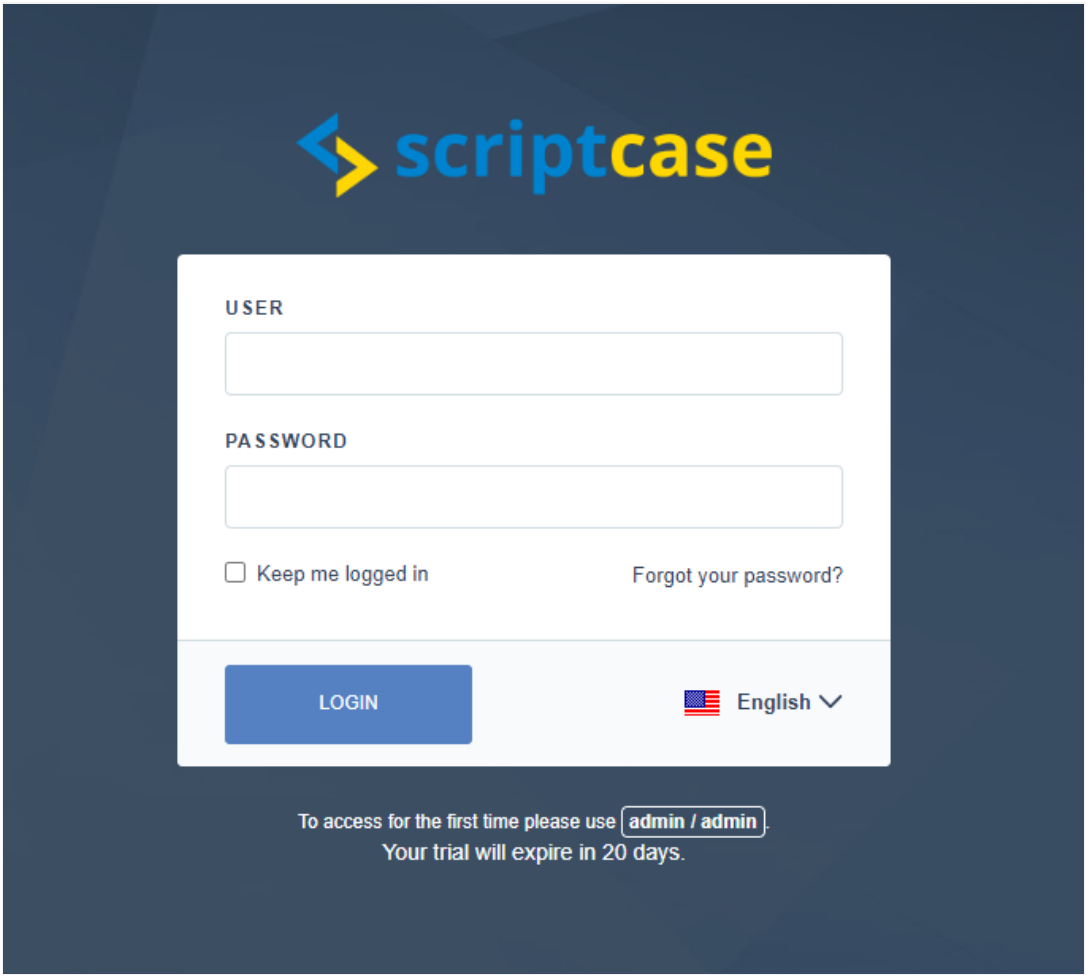
You must repeat the previously entered password for successful user creation.

Click proceed to proceed with the installation process.

5 - Complete Scriptcase custom installation on your web server.



After clicking proceed, you will be redirected to the Scriptcase Login page.



User

In this field, it is necessary to inform the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me connected

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

By clicking this option, it is possible to recover the password of the informed user. An email will be sent to the user in question so they can access the recovery steps.

Important: You can only perform password recovery if you have configured your SMTP information. To access our documentation showing how to perform this configuration click here.

Do you need to register your Scriptcase?

You can access our knowledge base to understand all the steps to register the Scriptcase serial.

Installing PHP 8.1 - macOS

This documentation will cover how installing PHP 8.1 and ScriptCase manually on macOS. **Performing this type of installation is your responsible for configuring your entire environment as well as database extensions.**

Listed below are versions of macOS approved for use by Scriptcase.

Compatibility Table

Scriptcase Automatic Installer is compatible with the following macOS versions

Version	Codename
macOS 15.0	“Sequoia”
macOS 14.2	“Sonoma”
macOS 13.0	“Ventura”
macOS 12.0	“Monterey”
macOS 11.0	“Big Sur”
macOS 10.15	“Catalina”
macOS v10.14	“Mojave”
macOS v10.13	“High Sierra”
macOS v10.12	“Sierra”
OS X v10.11	“El Capitan”

You can install the Scriptcase with PHP 7.3 or PHP 7.0:

See how to install with PHP 7.3
See how to install with PHP 7.0

Requirements

If you already have a Web Server with Apache and PHP 8.1 configured, just go to the SourceGuardia configuration and proceed to manual installation.

To proceed with the installation, you need to download some files.

Required Files:

- SourceGuardian Loader 13.0.3 for macOS - **macOS**: Click here
- Scriptcase **(.zip)**: Click here

Before proceeding with the installation it is recommended not to have any other web environments (such as XAMP, Zend Server, etc.) installed on the operating system. **Knowledge of Unix environment is recommended to use terminal.**

PHP configuration

Listed below are the commands required for the manual installation of PHP 8.1 on macOS. Check the package naming on your macOS terminal correctly.

1 - You'll need to install **Homebrew** to be able to access PHP 8.1 and other packages. Install **Brew** with the following commands in the macOS terminal:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

2 - In the macOS terminal, execute `brew update` to update **Brew**.

3 - Use the command `brew install shivammathur/php/php@8.1` for installing PHP 8.1.

4 - After the installation proccess completes, execute `brew info php@8.1` to get the information that we need to proceed with environment setting up.

```
php: stable 8.1.9 (bottled), HEAD
General-purpose scripting language
https://www.php.net/
/opt/homebrew/Cellar/php/8.1.9 (513 files, 81.7MB) *
  Poured from bottle on 2022-08-17 at 09:01:50
From: https://github.com/Homebrew/homebrew-core/blob/HEAD/Formula/php.rb
License: PHP-3.01
==> Dependencies
Build: httpd ✘, pkg-config ✘
Required: apr ✔, apr-util ✔, argon2 ✔, aspell ✔, autoconf ✔, curl ✔, freetds ✔, gd ✔, gettext ✔, gmp ✔, icu4c ✔,
krb5 ✔, libpq ✔, libsodium ✔, libzip ✔, oniguruma ✔, openldap ✔, openssl@1.1 ✔, pcre2 ✔, sqlite ✔, tidy-html5 ✔,
unixodbc ✔
==> Options
--HEAD
        Install HEAD version
==> Caveats
To enable PHP in Apache add the following to httpd.conf and restart Apache:
    LoadModule php_module /opt/homebrew/opt/php/lib/httpd/modules/libphp.so

    <FilesMatch \.php$>
        SetHandler application/x-httpd-php
    </FilesMatch>

Finally, check DirectoryIndex includes index.php
    DirectoryIndex index.php index.html

The php.ini and php-fpm.ini file can be found in:
    /opt/homebrew/etc/php/8.1/
```

5 - You need to configure Apache so that PHP 8.1 that was installed in the previous steps will be used as default on your system.

To use PHP on MacOS native Apache, we will need to create a certificate that authorizes its use. Until Monterey this process was optional, but now PHP Code Signed is required. Follow this post to do the code signing

5.1 - Edit the Apache file `httpd.conf` , find the "**Loadmodule**" line block and add the line according to your equipment:

Exemple:

```
sudo nano /etc/apache2/httpd.conf
```

Chip Intel

```
LoadModule php_module /usr/local/opt/php@8.1/lib/httpd/modules/libphp.so
```

Chip da Apple

```
LoadModule php_module/opt/homebrew/opt/php@8.1/lib/httpd/modules/libphp.so
```

At the end of the file add:

```
<FilesMatch .php$>
SetHandler application/x-httpd-php
</FilesMatch>
```

6 - Add path to PHP 8.1 module installed after last line `LoadModule` in **httpd.conf**.

Exemple:

`LoadModule php_module /usr/local/opt/php@8.1/lib/httpd/modules/libphp.so`

```
#LoadModule rewrite_module libexec/apache2/mod_rewrite.so
#PHP was deprecated in macOS 11 and removed from macOS 12
#LoadModule perl_module libexec/apache2/mod_perl.so
LoadModule hfs_apple_module libexec/apache2/mod_hfs_apple.so
LoadModule php_module /opt/homebrew/opt/php@8.1/lib/httpd/modules/libphp.so "PHP 81"
```

`

After performing this action, save the file and restart the apache service.

`sudo apachectl restart`

7 - Add the path to the installation of PHP 8.1 in the `~ /.zshrc` file with the following commands:

`echo 'export PATH="/opt/homebrew/opt/php@8.1/bin:$PATH"' >> ~/.zshrc` `echo 'exportPATH="/opt/homebrew/opt/php@8.1/sbin:$PATH"' >> ~/.zshrc`

8 - Edit the file `~/.bash_profile` and add PHP 8.1 path with these following commands:

`echo 'export PATH="/opt/homebrew/opt/php@8.1/bin:$PATH"'`

9 - Verify PHP version by running command: `php -v`

Exemple:

```
qascriptcase@MacBook-Air-de-scriptcase ~ % php -v
PHP 8.1.9 (cli) (built: Aug  4 2022 05:22:38) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.9, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.9, Copyright (c), by Zend Technologies
```

10 - Access the **php.ini** file in the directory. `/opt/homebrew/etc/php/8.1/` and set the recommended minimum value correctly of these PHP directives listed below for Scriptcase work properly.

Search for the directives and assign the value according to this example:

- `max_execution_time = 3600`
- `max_input_time = 3600`
- `max_input_vars = 10000`
- `memory_limit = 1024M`
- `post_max_size = 1024M`
- `upload_max_filesize = 1024M`
- `max_file_uploads = 200`
- `short_open_tag = On`

11 - By default, PHP disables some functions that are important to Scriptcase. Click here to access the necessary functions list for ScriptCase

- Change item line **disable_functions** as example below:

`disable_functions =`

Exemple:

```
; This directive allows you to disable certain functions for security reasons.
; It receives a comma-delimited list of function names.
; http://php.net/disable-functions
disable_functions =
```

12 - Set PHP **TimeZone** according to your region. Use the value available in the PHP documentation. Search for **date.timezone** in your **php.ini** file, uncomment it and edit according to the selected TimeZone.

Exemple:

`date.timezone = America/Recife`

13 - Set the folder where temporary files will be stored. search for `;session.save_path` , uncomment it and inform your temporary folder path.

Exemple:

`session.save_path = "/tmp"`

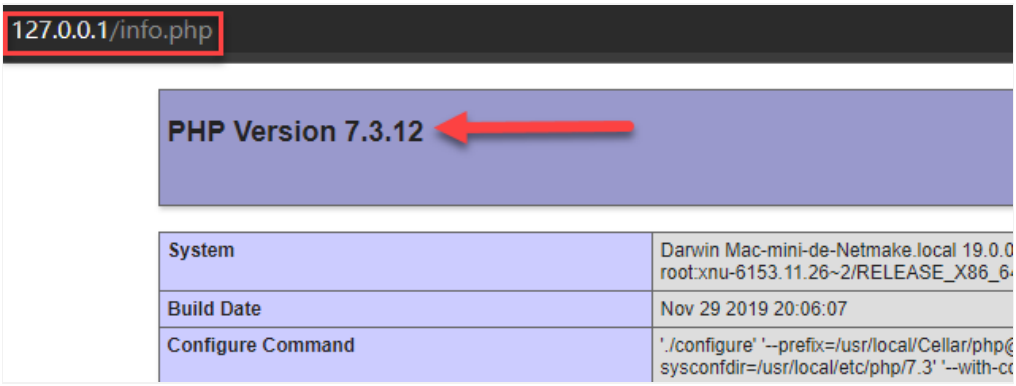
14 - Save all modifications made and restart the Apache service using the following command in terminal:

`sudo apachectl restart`

15 - Verify changes made through the file **info.php**. You need to create this file and put it in the directory. `/Library/WebServer/Documents/` with the following content:

```
<?php
phpinfo();
?>
```

After that, check in your browser the page generated by accessing the URL `127.0.0.1/info.php` .



NOTE: For more information about the function **phpinfo()**, verify our PHP documentation.

Enable SourceGuardian Loader

Before you start installing Scriptcase, you need to enable the loader responsible for encrypting Scriptcase in PHP. Follow the steps below to perform this action successfully.

1 - Access the Downloads folder and extract the downloaded file referring to **SourceGuardian Loader** at the beginning of this documentation.

If you have not yet downloaded the SourceGuardian Loader click here and download the file according to the architecture used.

Exemple:

`cd /Users/usuario/Downloads`

`sudo chmod 777 loaders.macosx-arm64.zip && unzip loaders.macosx-arm64.zip`

`cd loaders.macosx-arm64`

2 - Copy the file **ixed.8.1.dar** and paste it into your PHP **extension_dir** directory. We'll use `/opt/homebrew/lib/php/pecl/20210902` for this example.

Exemple:

`sudo cp ixed.8.1.dar /opt/homebrew/lib/php/pecl/20210902`

3 - Edit the **php.ini** file and below the last line enter the path to the extension within the parameter `zend_extension` :

Exemple:

`sudo bash -c 'echo "zend_extension=/opt/homebrew/lib/php/pecl/20210902/ixed.8.1.dar" > /opt/homebrew/etc/php/8.1/conf.d/sourceguardian.ini'`

4 - Restart the Apache service with the following terminal command:

`sudo apachectl restart`

Manual ScriptCase installation

Listed below are the steps needed to do a Scriptcase manual installation. **To do the steps, it is needed that you have a web server configured in your machine**.

1 - Download the Scriptcase **(.zip)** directly from the download page from our website.

2 - Extract the **.zip** file and rename the extracted folder to `scriptcase` .

3 - Move the **scriptcase** to your server root. **Depending of your operating system and the installation the path can be different from the example.** However, by default the paths are:

Windows	Server
`C:\Apache24\htdocs\`	Apache
`C:\inetpub\wwwroot\`	IIS

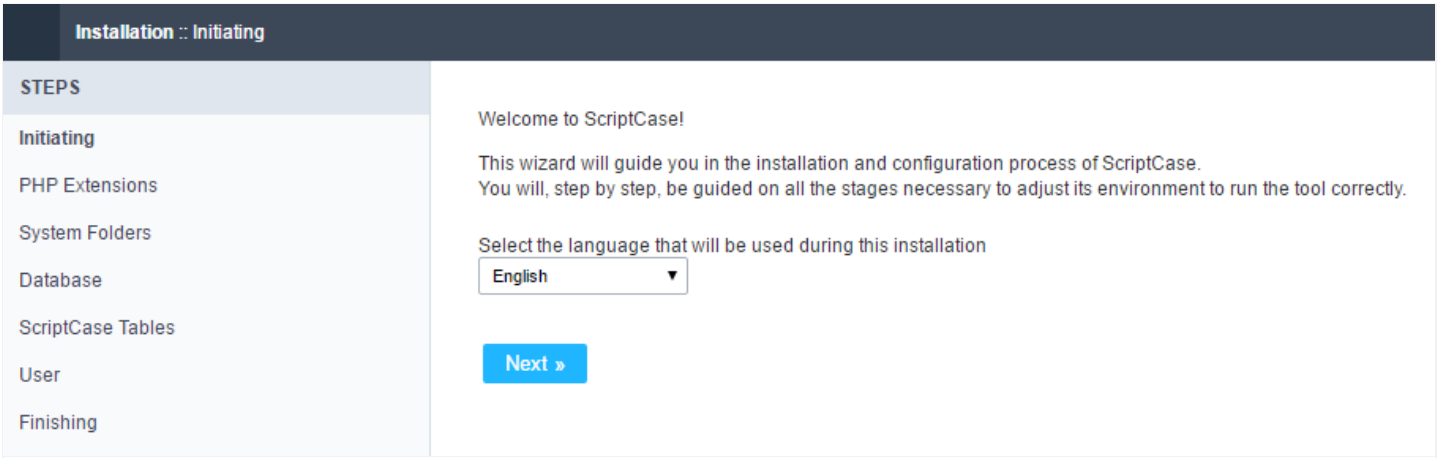
Linux	Server
`/var/www/html/`	Linux Local
`/home/$(whoami)/public_html/`	Linux Server

macOS	Server
`/Library/WebServer/Documents`	Apache

4 - Access the Scriptcase using your browser:

`127.0.0.1/scriptcase`

- **By default the selected language is English.**

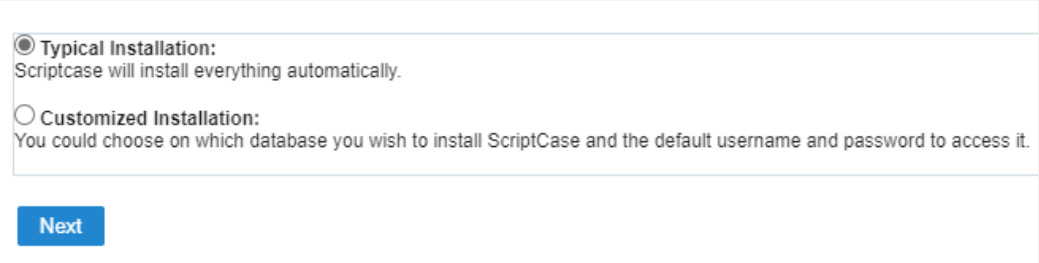


5 - Check the extensions required for Scriptcase to function and database connection modules that are enabled.

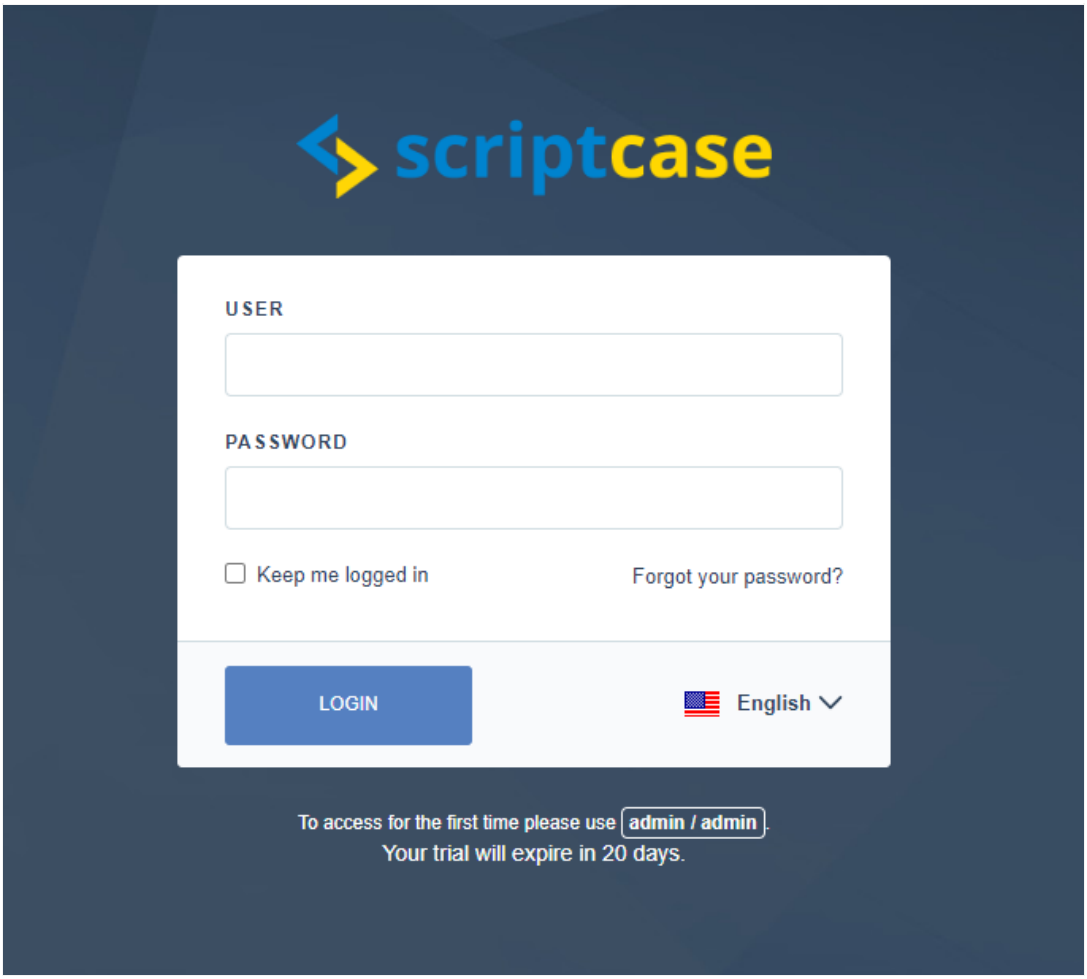
Typical installation

Check the steps below to complete the manual installation of Scriptcase in a typical way.

NOTE: By selecting this installation, Scriptcase will set your environment by default. Defining the installation database and the login user in Scriptcase.



After clicking proceed, you will be redirected to the Scriptcase Login page.



User

In this field, it is necessary to inform the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me connected

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

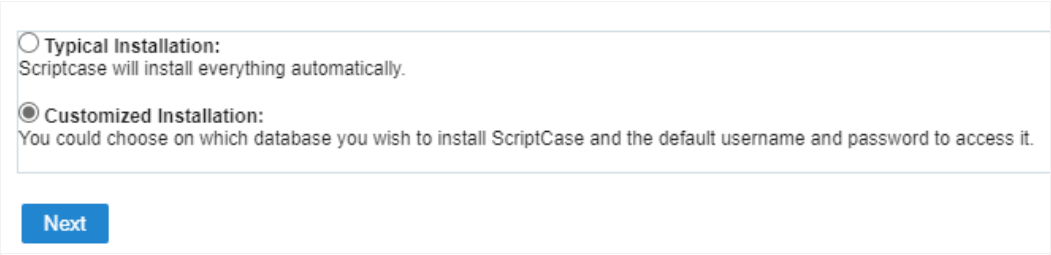
By clicking this option, it is possible to recover the password of the informed user. An email will be sent to the user in question so they can access the recovery steps.

Important: You can only perform password recovery if you have configured your SMTP information. To access our documentation showing how to perform this configuration click here.

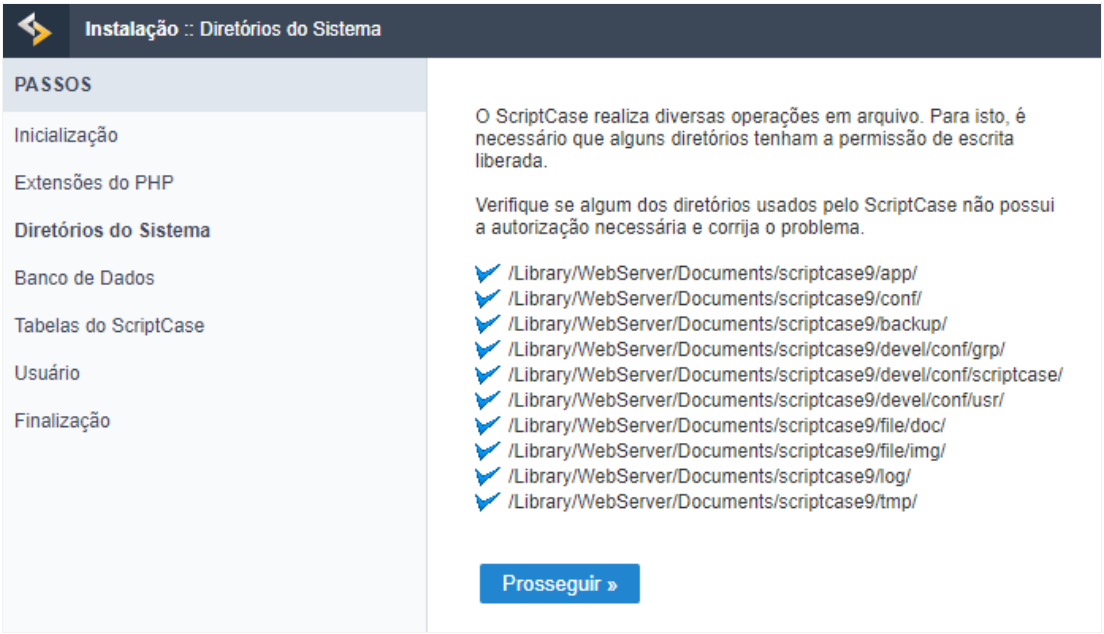
Custom installation

Check the steps below to complete the manual installation of Scriptcase in a custom manner.

NOTE: By selecting this installation, you will be responsible for setting up your environment. Defining the installation database and the login user for Scriptcase.

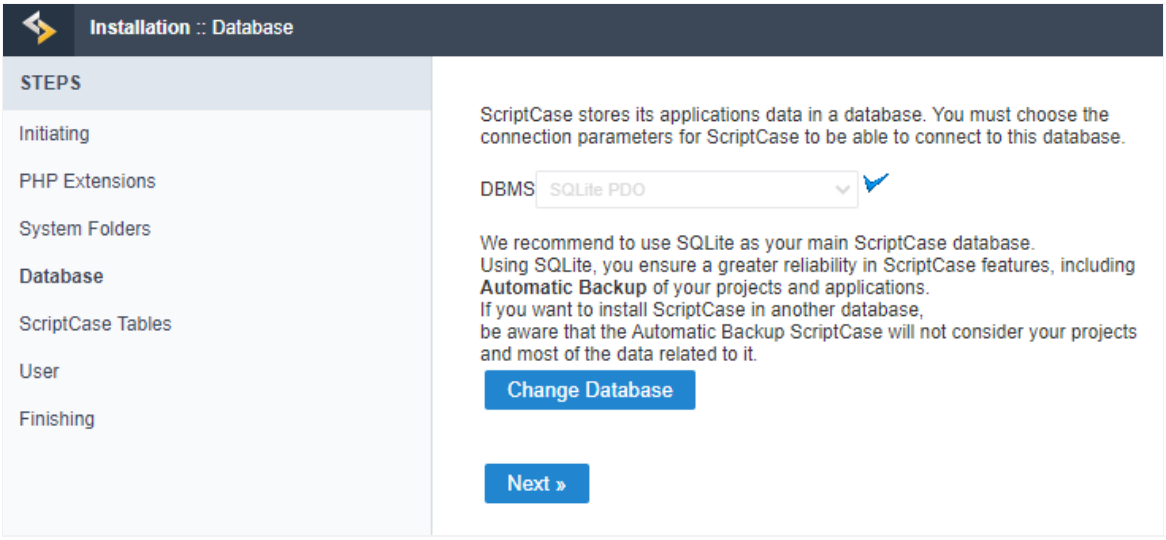


1- Verify that the directories required by Scriptcase have the correct permissions. If any directory listed does not have the active check icon , correct permission for this directory.

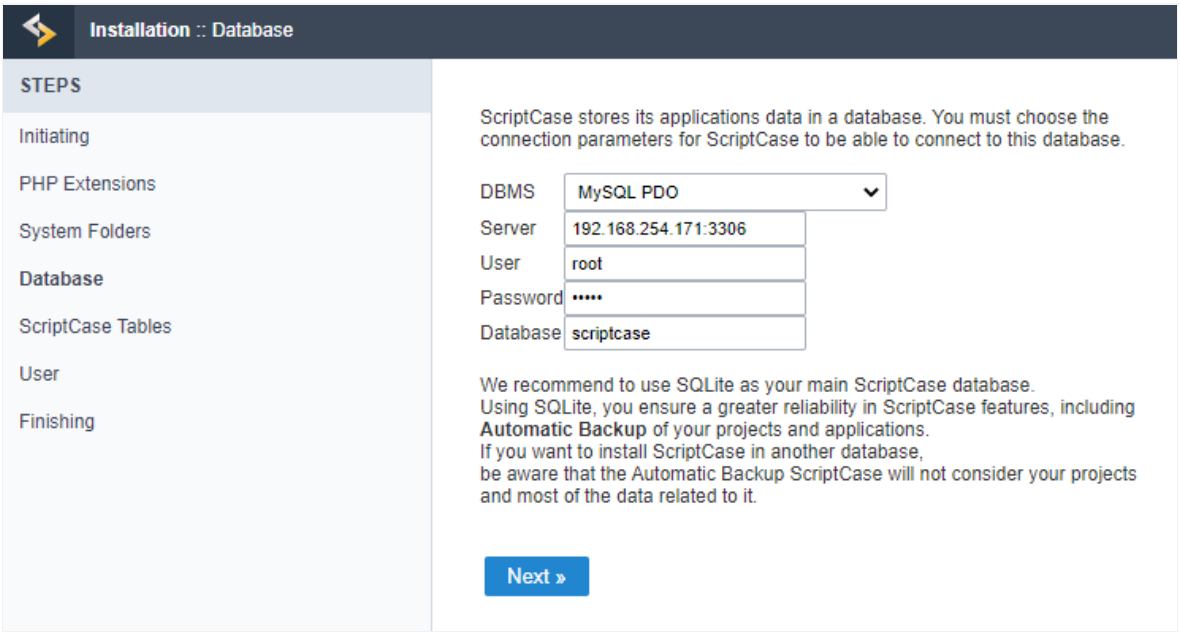


Clique em prosseguir para avançar no processo de instalação.

2- Choose the database where the Scriptcase database will be installed. By default, Scriptcase recommends that it be installed with the **SQLite** database.



- You can change the default database by clicking the **Change Database** button. By clicking this option you will need to set the following options:



DBMS

In this option, you will select the Driver to connect to your database.

Server

In this option, you will inform the server to connect to the desired database.

User

You need to inform a user who has access to the database you want Scriptcase to be installed on.

Password

You must enter the authentication password corresponding to the entered user.

Database

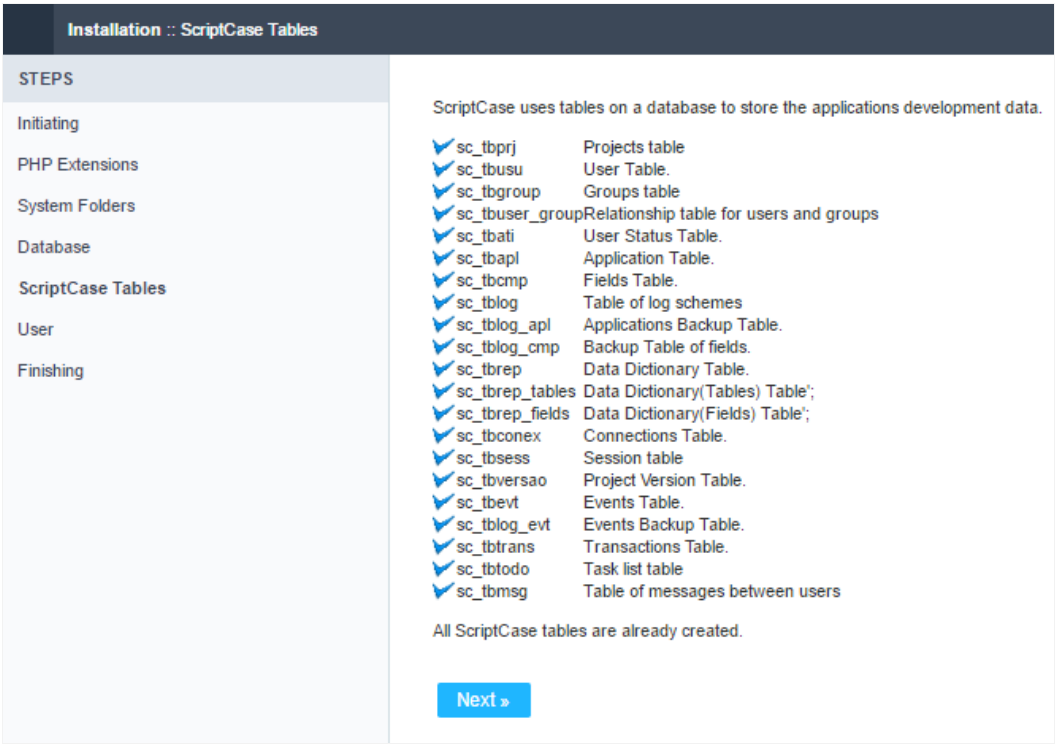
You must enter the name of the database where Scriptcase will be installed.

Example:

`scriptcase`

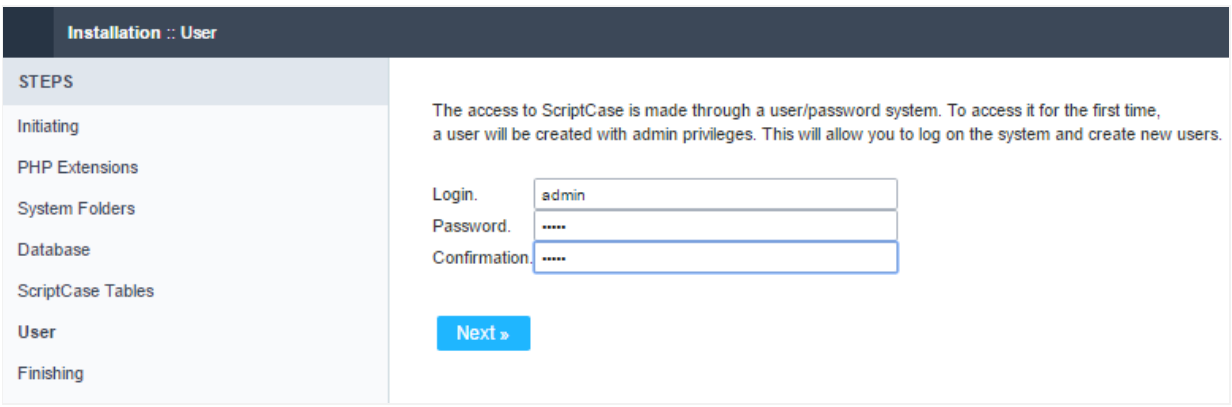
Clique em prosseguir para avançar no processo de instalação.

3 - Verifique se a criação das tabelas da base de dados do Scriptcase ocorreu com sucesso. As tabelas serão marcadas com o ícone de sucesso na lateral.



Click proceed to proceed with the installation process.

_4 -__ Set the default user who will have access to Scriptcase.



Login

In this option, you will enter the default Scriptcase User Login.

Password

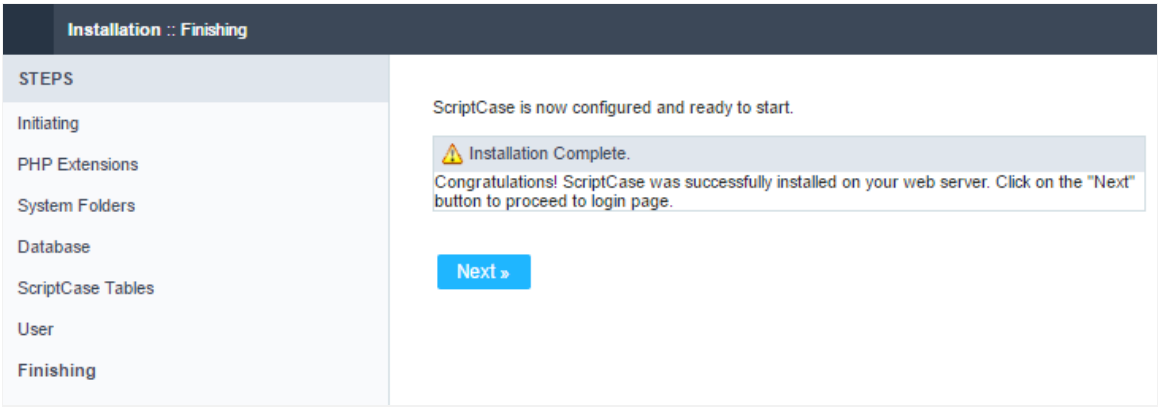
In this option, you must enter the password that will correspond to the user informed.

Confirmation

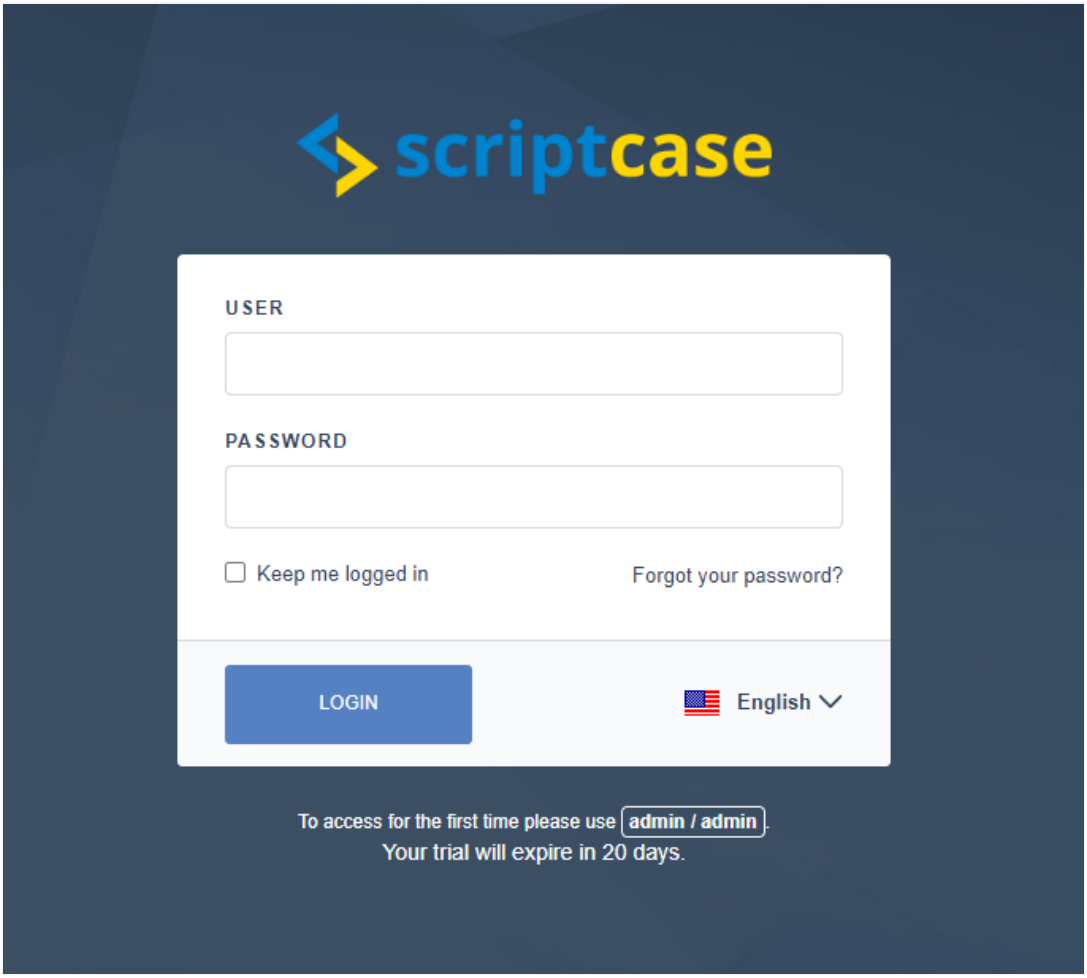
You must repeat the previously entered password for successful user creation.

Click proceed to proceed with the installation process.

5 - Complete Scriptcase custom installation on your web server.



After clicking proceed, you will be redirected to the Scriptcase Login page.



User

In this field, it is necessary to inform the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me connected

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

By clicking this option, it is possible to recover the password of the informed user. An email will be sent to the user in question so they can access the recovery steps.

Important: You can only perform password recovery if you have configured your SMTP information. To access our documentation showing how to perform this configuration click here.

Needing register your ScriptCase?

VYou can access our knowledge base to understand all the steps for registering Scriptcase license.

Scriptcase Manual Installation

Listed below are the steps needed to do a Scriptcase manual installation. **To do the steps, it is needed that you have a web server configured in your machine**.

1 - Download the Scriptcase **(.zip)** directly from the download page from our website.

2 - Extract the **.zip** file and rename the extracted folder to `scriptcase` .

3 - Move the **scriptcase** to your server root. **Depending of your operating system and the installation the path can be different from the example.** However, by default the paths are:

Windows	Server
`C:\Apache24\htdocs\`	Apache
`C:\inetpub\wwwroot\`	IIS

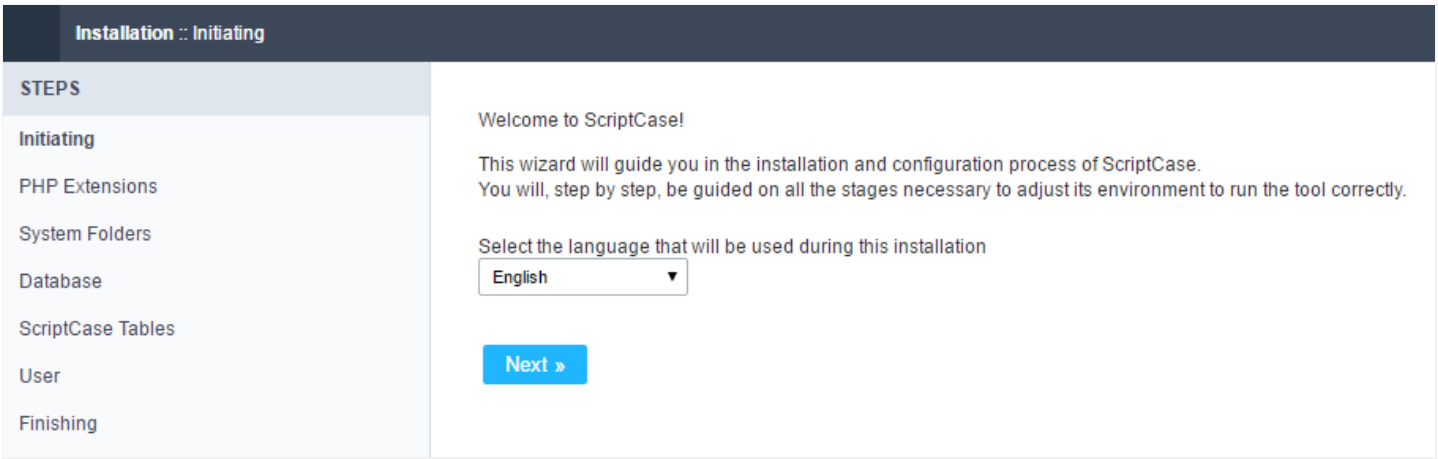
Linux	Server
`/var/www/html/`	Linux Local
`/home/$(whoami)/public_html/`	Linux Server

macOS	Server
`/Library/WebServer/Documents`	Apache

4 - Access the Scriptcase using your browser:

`127.0.0.1/scriptcase`

- **By default the selected language is English.**



5 - Check the extensions required for Scriptcase to function and database connection modules that are enabled.

Check if the necessary PHP extensions are loaded.

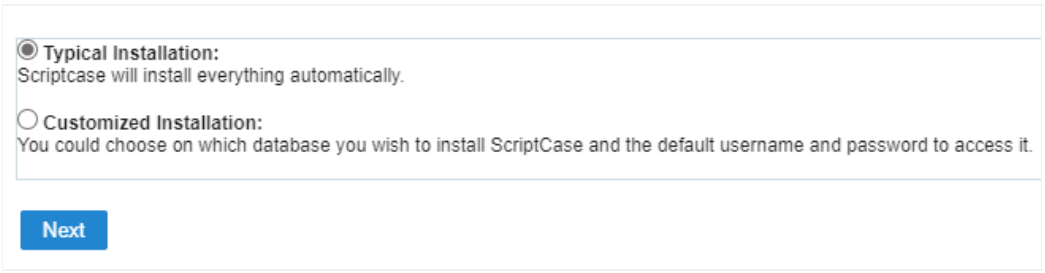
Required Extensions.		Action.
GD	This extension is used to create charts and manipulate images	
MBSTRING	This extension is used to convert special chars	
SimpleXML	The SimpleXML extension provides a very simple and easily usable toolset to convert XML to an object that can be processed with normal property selectors and array iterators.	
ZIP	This extension is used to create and extract files from a ZIP archive	
ZLIB	This extension is used to extract files from a ZIP archive	
JSON	This extension implements the data interchange format JavaScript Object Notation (JSON).	

Database Extensions.		Action.
com_dotnet	Generic ADO, MS Access ADO, MS SQL Server ADO	
ibm_db2	DB2, DB2 ODBC NATIVE	Enable.
interbase	Firebird, Interbase 6, Interbase 6.5	
mssql	MS SQL Server 7	Enable.
mysql	MaxSQL, MySQL (Non-Transactional), MySQL (Transactional)	Enable.
mysqli	MySQLi	
oci8	Oracle 8, Oracle 8 Portable, Oracle 8.0.5	
odbc	DB2 ODBC GENERIC, DB2 ODBC GENERIC 6, Generic ODBC, MS Access ODBC, MS SQL Server ODBC, Oracle ODBC, Progress	
oracle	Oracle 7	Enable.
pdo_dblib	Sybase PDO DBLIB	
pdo_firebird	Firebird PDO	
pdo_ibm	PDO IBM	Enable.
pdo_informix	Informix PDO	Enable.

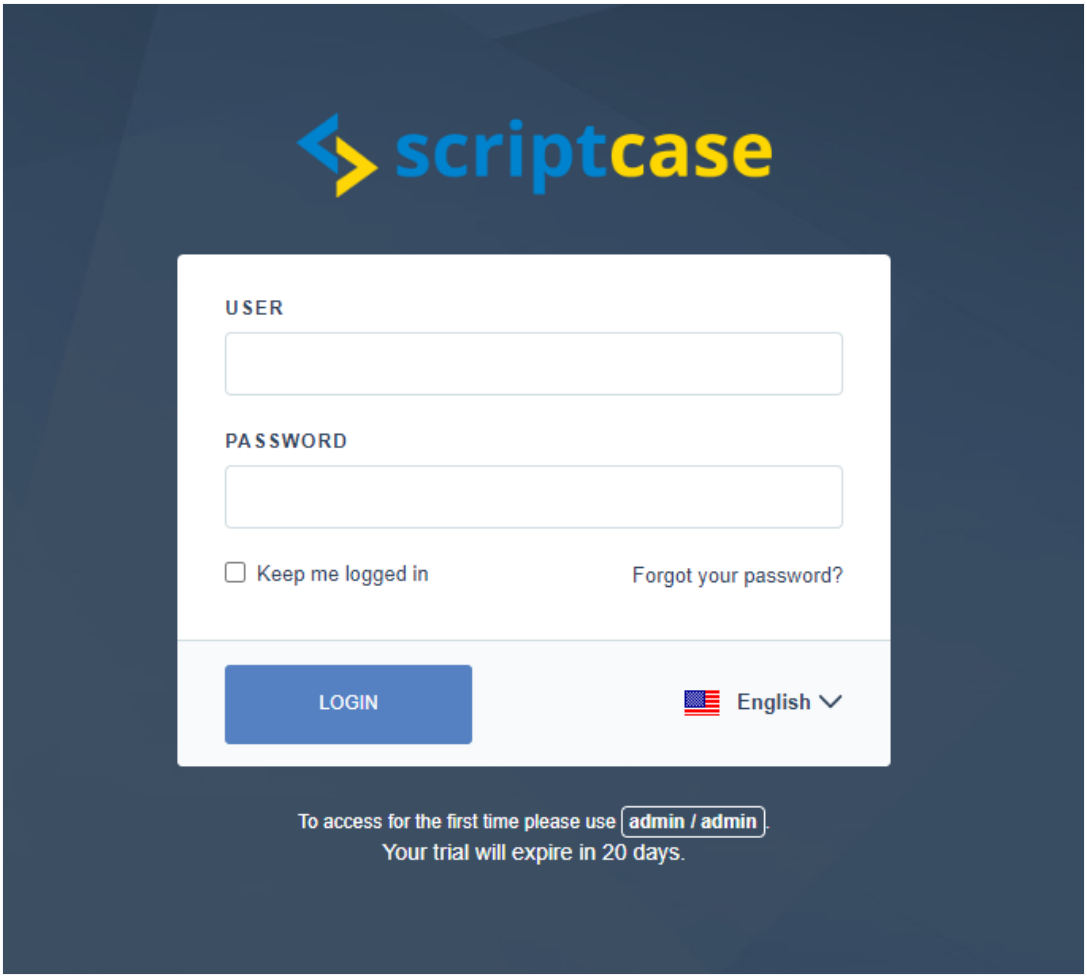
Typical Installation

Check the steps below to complete the manual installation of Scriptcase in a typical way.

NOTE: By selecting this installation, Scriptcase will set your environment by default. Defining the installation database and the login user in Scriptcase.



After clicking proceed, you will be redirected to the Scriptcase Login page.



User

In this field, it is necessary to inform the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me connected

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

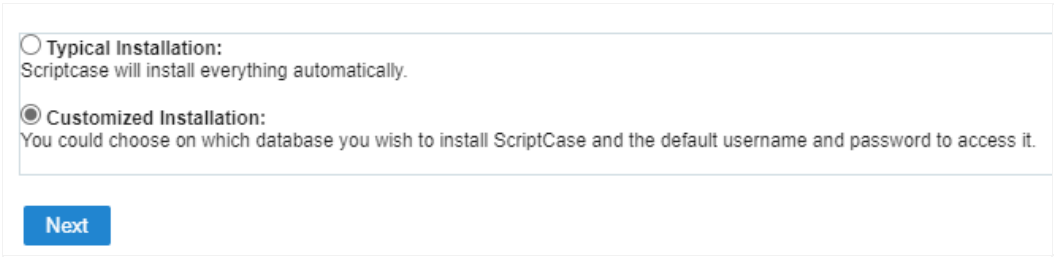
By clicking this option, it is possible to recover the password of the informed user. An email will be sent to the user in question so they can access the recovery steps.

Important: You can only perform password recovery if you have configured your SMTP information. To access our documentation showing how to perform this configuration click here.

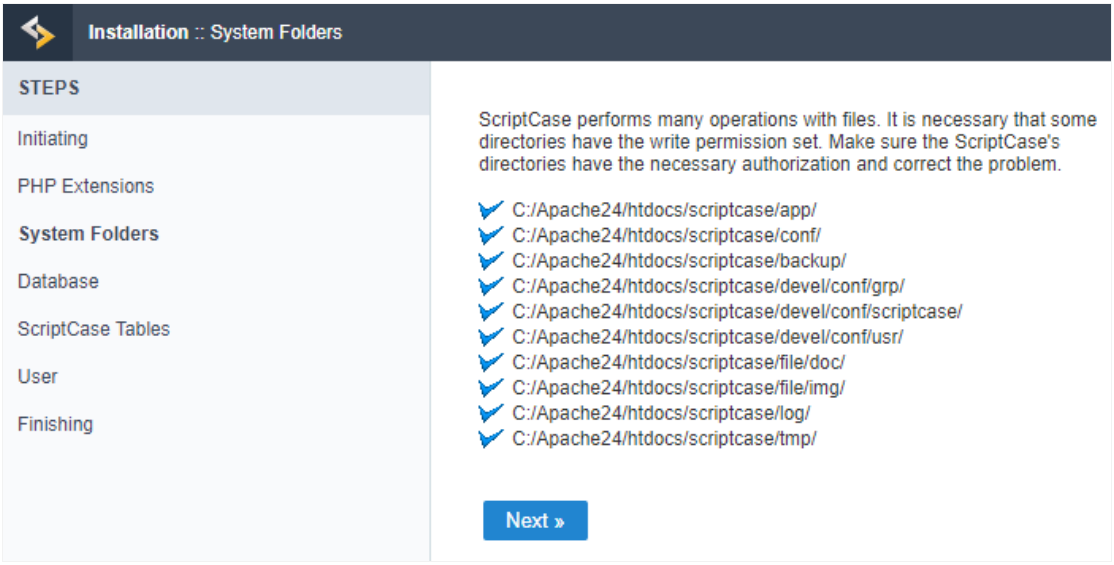
Custom Installation

Check the steps below to complete the manual installation of Scriptcase in a custom manner.

NOTE: By selecting this installation, you will be responsible for setting up your environment. Defining the installation database and the login user for Scriptcase.

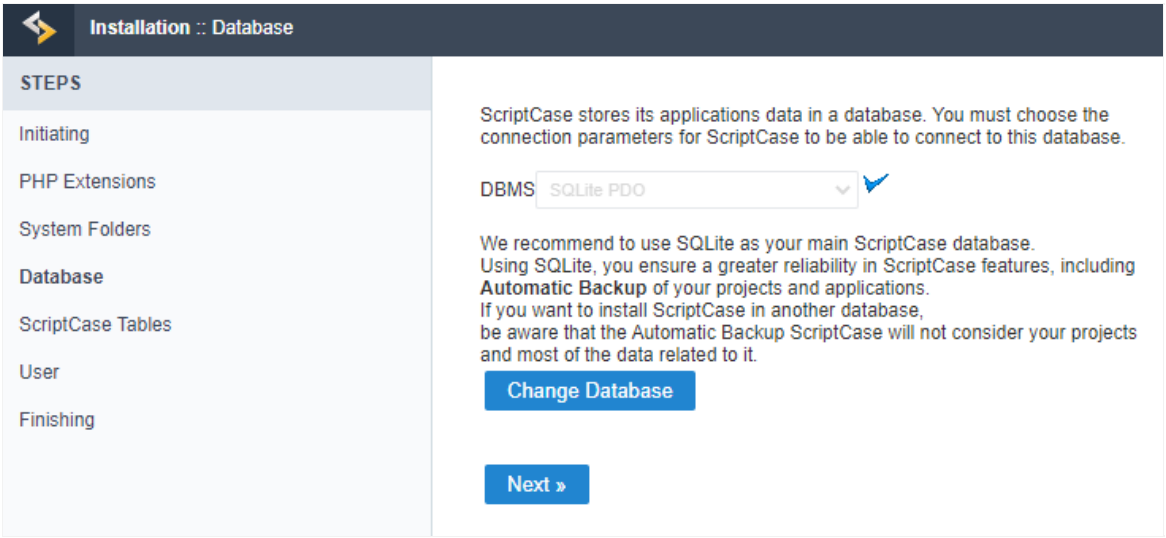


1- Verify that the directories required by Scriptcase have the correct permissions. If any directory listed does not have the active check icon , correct permission for this directory.

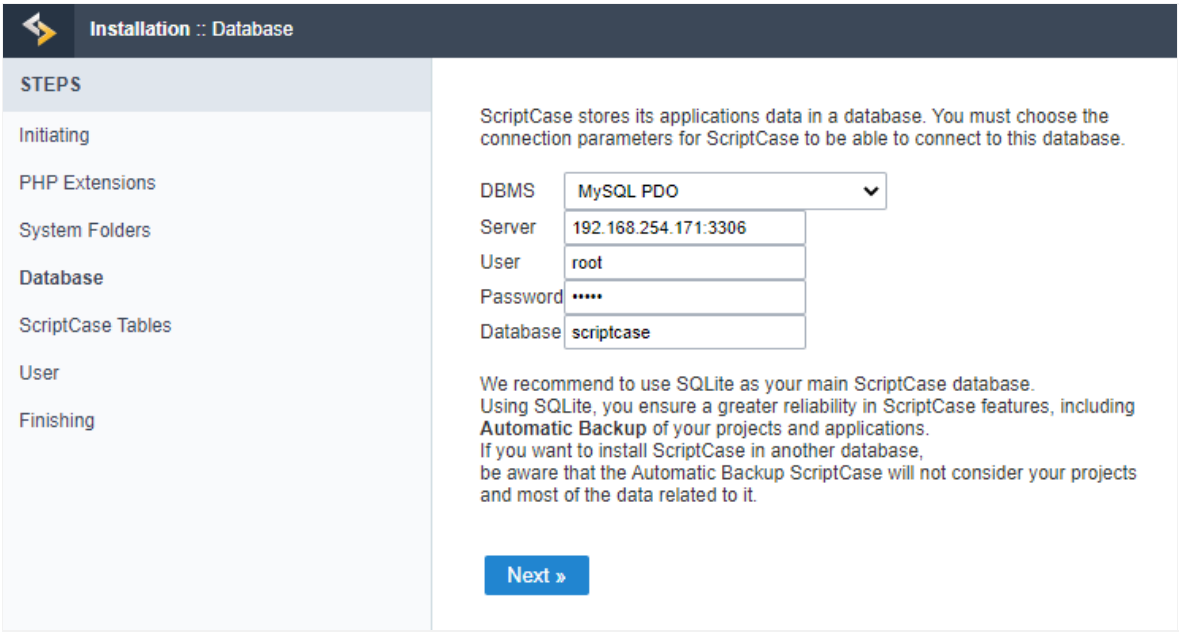


Click proceed to proceed with the installation process .

2- Choose the database where the Scriptcase database will be installed. By default, Scriptcase recommends that it be installed with the **SQLite** database.



- You can change the default database by clicking the **Change Database** button. By clicking this option you will need to set the following options:



DBMS

In this option, you will select the Driver to connect to your database.

Server

In this option, you will inform the server to connect to the desired database.

User

You need to inform a user who has access to the database you want Scriptcase to be installed on.

Password

You must enter the authentication password corresponding to the entered user.

Database

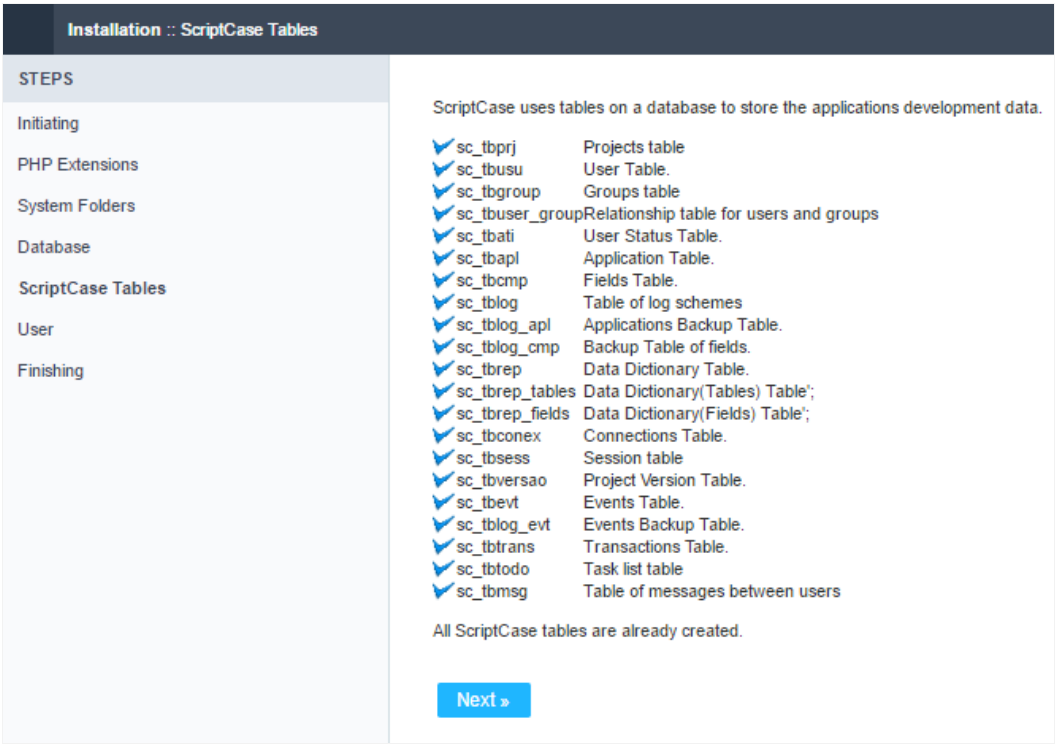
You must enter the name of the database where Scriptcase will be installed.

Example:

`scriptcase`

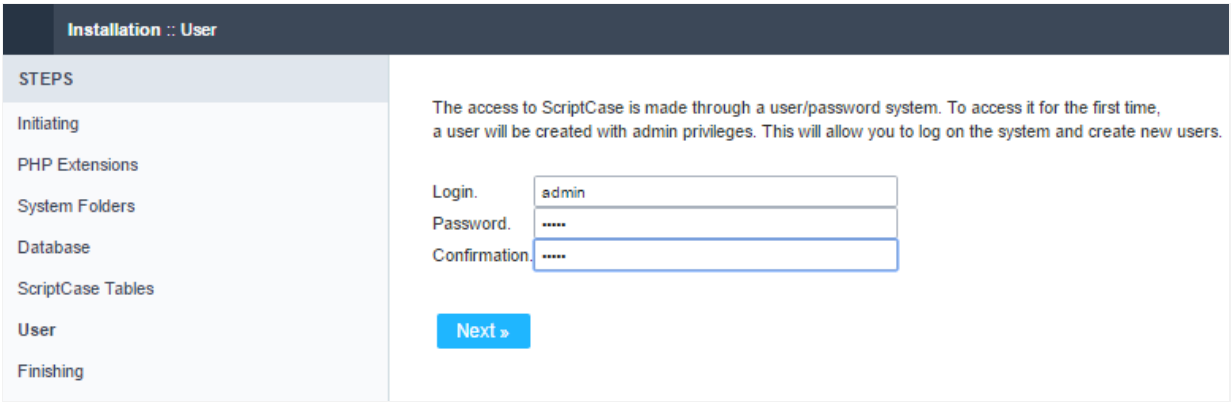
Clique em prosseguir para avançar no processo de instalação.

3 - Verifique se a criação das tabelas da base de dados do Scriptcase ocorreu com sucesso. As tabelas serão marcadas com o ícone de sucesso na lateral.



Click proceed to proceed with the installation process.

_4 -__ Set the default user who will have access to Scriptcase.



Login

In this option, you will enter the default Scriptcase User Login.

Password

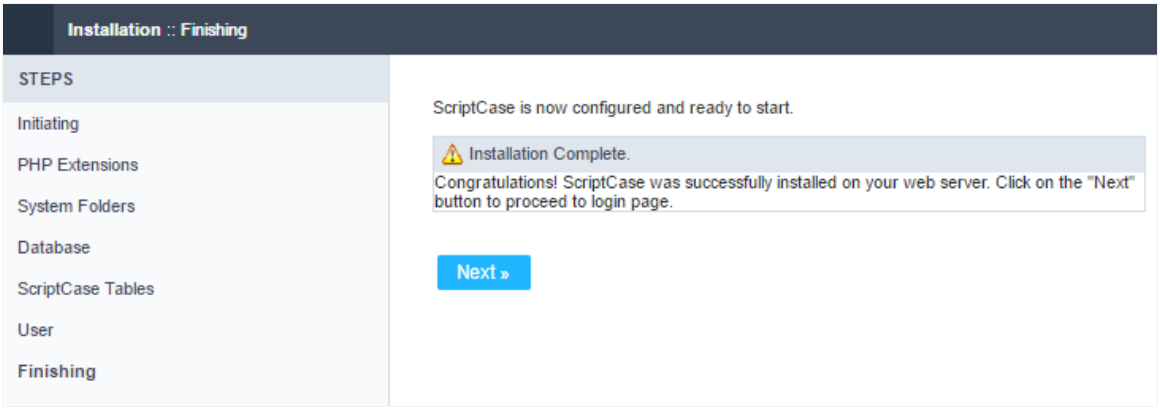
In this option, you must enter the password that will correspond to the user informed.

Confirmation

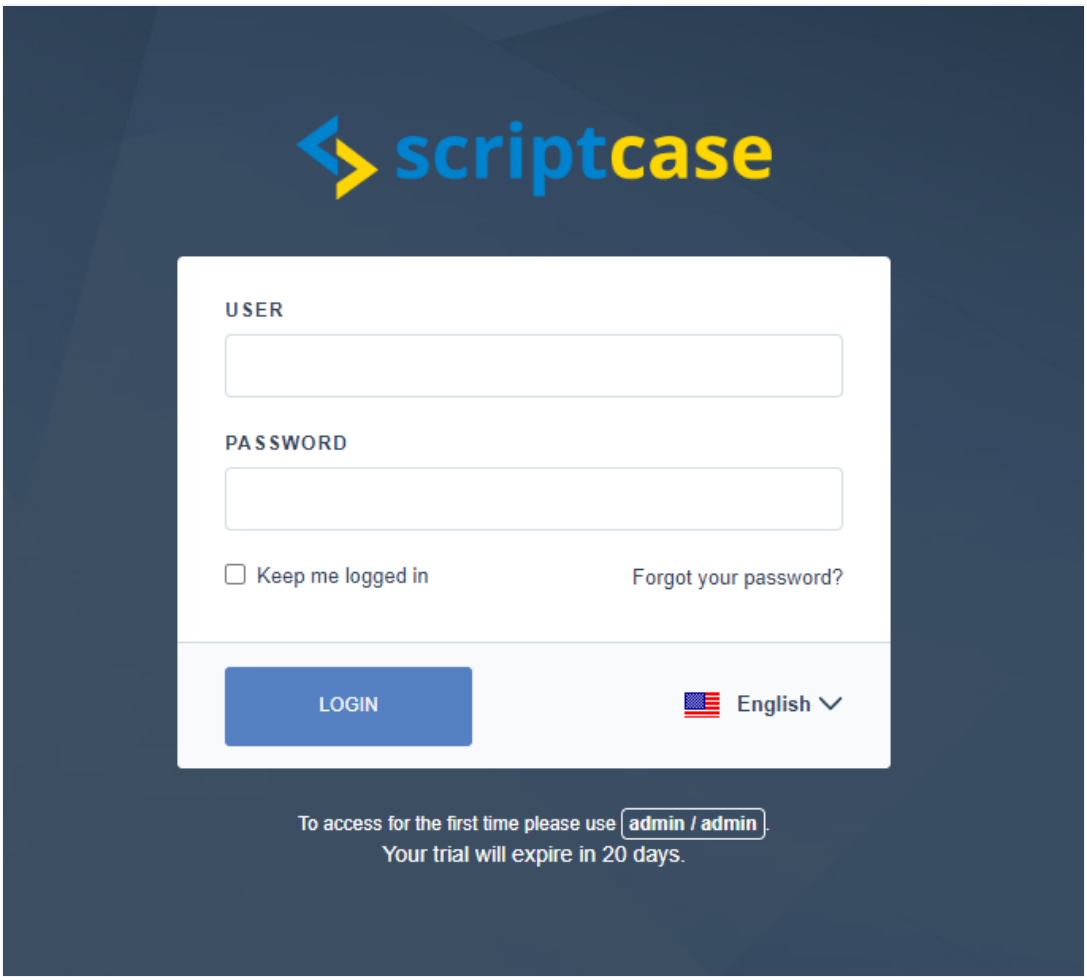
You must repeat the previously entered password for successful user creation.

Click proceed to proceed with the installation process.

5 - Complete Scriptcase custom installation on your web server.



After clicking proceed, you will be redirected to the Scriptcase Login page.



User

In this field, it is necessary to inform the user to login to Scriptcase. By default, the user is **admin**.

Password

In this field, you must enter the password to login to Scriptcase. By default, the password is **admin**.

Keep me connected

This option keeps the user logged in until they log out through the Scriptcase interface.

Forgot your password?

By clicking this option, it is possible to recover the password of the informed user. An email will be sent to the user in question so they can access the recovery steps.

Important: You can only perform password recovery if you have configured your SMTP information. To access our documentation showing how to perform this configuration click here.

Need to register your Scriptcase?

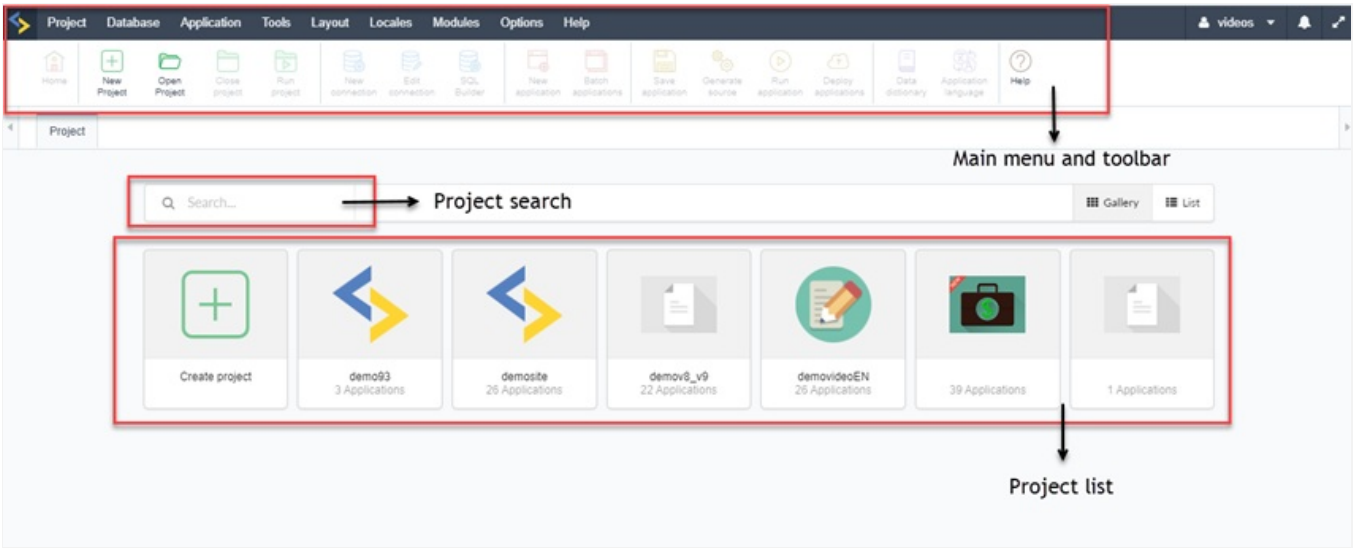
You can access our knowledge base to understand all the steps for registering a Scriptcase license.

Scriptcase Interface Overview

In this article, we present to you the main options from Scriptcase's interface. There are two important screens: project dashboard and applications management interface.

Project Dashboard

The project dash is the first interface you will see once you log into Scriptcase. It's where you start a new project or select an existing project for editing.



Main Menu

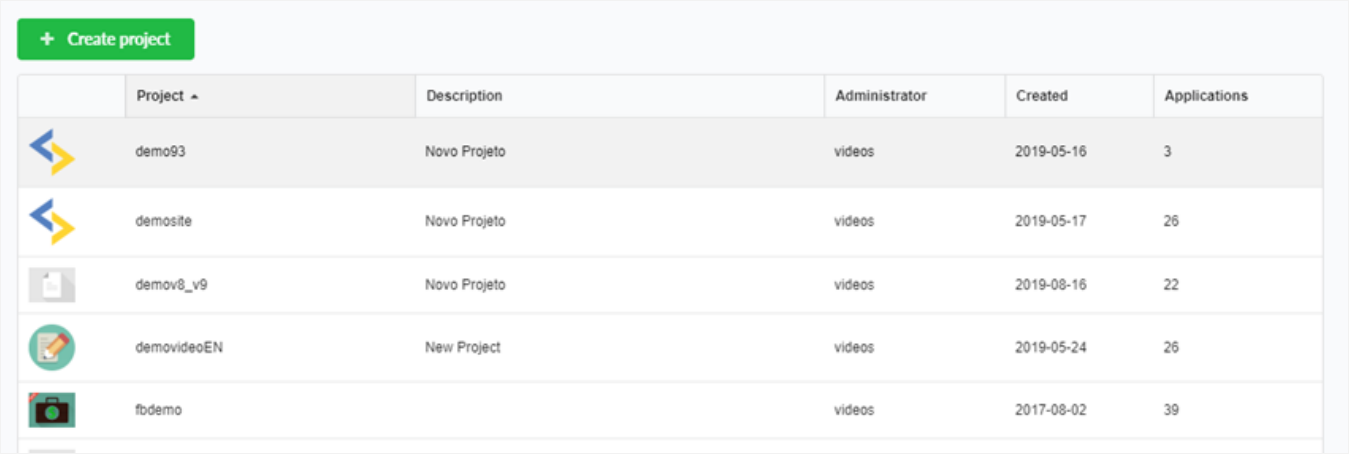
The main menu contains both Scriptcase's and project's general settings, and it is available at the top of the page. Most of the options are related to the project settings and are not available before the project creation.

Toolbar

The toolbar is displayed right below the main menu, where you can find the most frequent used functions used. The toolbar is customizable, and you can create new functions as well as edit the current ones. Learn more about the toolbar by clicking here.

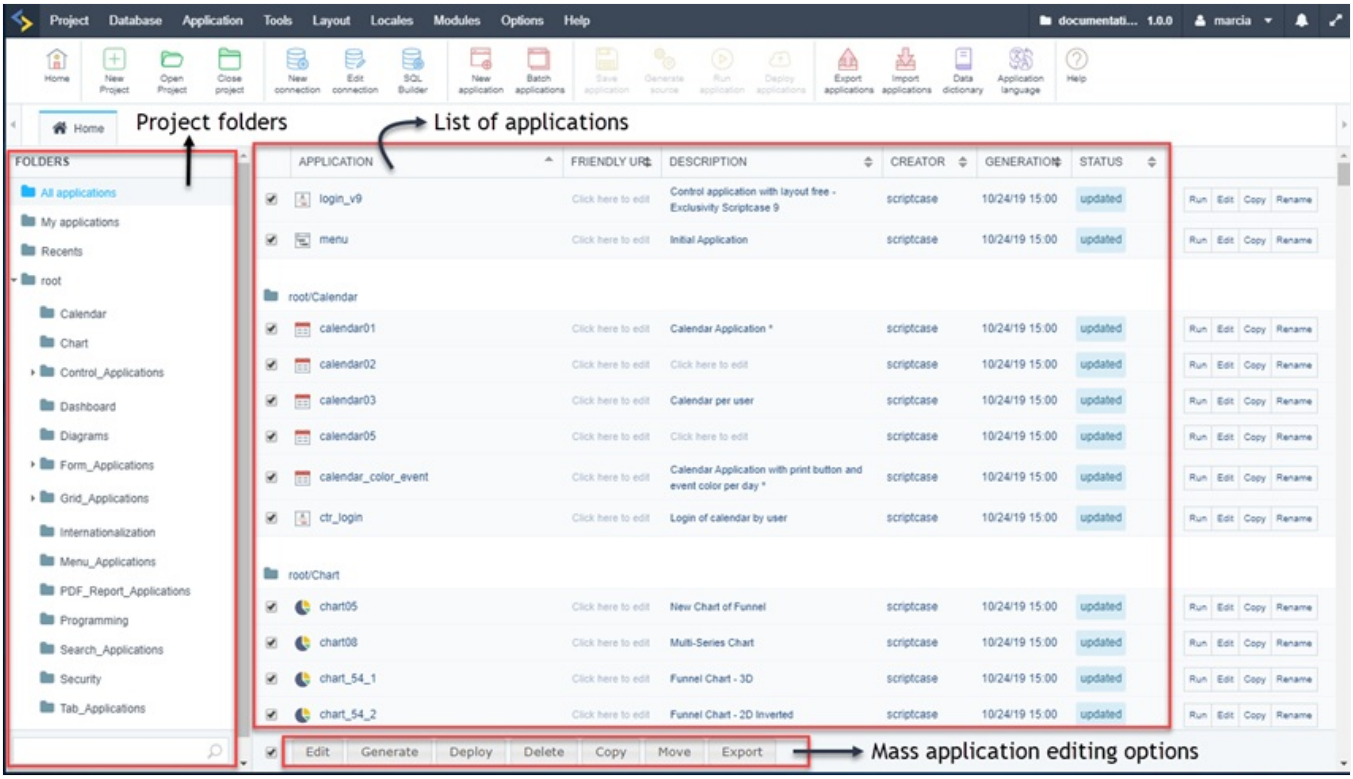
Project List

On this session is displayed a list with all the Scriptcase projects. You can display the project in a gallery or in list (to sort by date, name or creator). You have also a general search option.



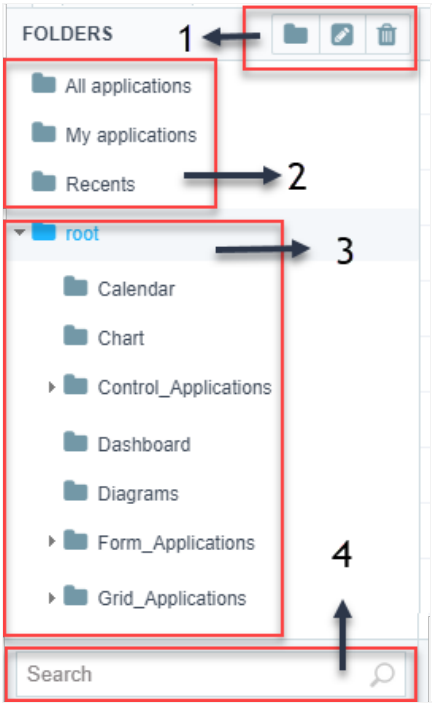
Application management interface

Once you create the project Scriptcase will display the applications management interface, where you can organize all project applications using folders. You are able also to search created applications, sort by date, creator, status, and take some actions such as run, generate, deploy, rename and copy. You can edit one or various applications at a time.



Project Explorer

In this session, you can find all the folders created to organize the applications. There's also the folder that contains all the applications of the project in one place.



- 1 - Folders functions;
- 2 - Filters;
- 3 - Folders' structure;
- 4 - Application search.

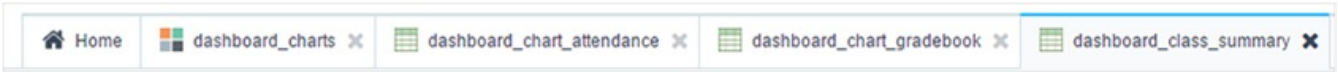
List of Applications

You can view a list of the applications that are in the selected folder in the Project Explorer presented on the left side.

- 1 - Columns;
- 2 - Express options;
- 3 - Footer Options.

Taskbar

This bar holds the tabs of all applications which are open for editing.



Main Menu

The main menu contains both Scriptcase's and project's general settings, and it is available at the top of the page. Most of the options are related to the project settings and are not available before the project creation. See below each item and a resume of its options:

Project

- **New Project**: To create a new project or import one of the system templates.
- **Open Project**: Lists the projects that the user is permitted to access.
- **Close Project**: Closes the current project.
- **Properties**: Allows you to edit some of the project properties such as themes and languages.
- **Default Values**: Allows editing the current project default values.
- **Version History**: Displays a list of project versions.
- **Increment Version**: This option helps you to make the project version control.
- **Export Project**: Allows to export one project at a time, taking all the project's essential content for using in another Scriptcase of the same version.
- **Import Project**: Allows to import a project exported by Scriptcase of the same version.
- **Generate Source Code**: Generates the source code for all applications in the current project.
- **Deploy Applications**: Opens the Publishing Wizard for publishing the applications of the current project.
- **Delete Project**: This option is only available once you are into a project. This option deletes a project.
- **Reports**:
  - **Application List**: Displays the project applications with details of its creation.
  - **Developers Summary**: Displays a summary of the applications and codes created by a developer.
- **Show Diagram**: Displays a complete diagram of the project showing the relationship between the applications.
- **Project Search**: It allows to perform a search for texts in all the project's applications.

Database

- **New Connection**: Allows you to create a connection in the current project.
- **Edit Connection**: Allows you to edit existing connections in the current project.
- **Import ACCESS**: Allows the conversion of an ACCESS data source into a structured database (MySQL, PostgreSQL, SQLite, and SQL Server).
- **Import EXCEL**: Allows the conversion of an EXCEL data source into a structured database (MySQL, PostgreSQL, SQLite, and SQL Server).
- **Import CSV**: Allows the conversion of a CSV data source into a structured database (MySQL, PostgreSQL, SQLite, and SQL Server).
- **SQL Builder**: Tool for creating and executing SQL commands.
- **Database Builder**: Tool for rapidly creating and editing tables in the database.
- **ER Diagram**: Tool that allows the visualization of the relationships between tables of the connected database.

Application

- **New Application**: Create a new application.
- **Batch Applications**: Allows simultaneous creation of Grids and Forms.
- **Restore Applications**: Allows the restoration of applications, where is displayed a list of the last edited applications.
- **Save Application**: Save changes made to the current application.
- **Generate Source**: Generate the Source Code for all the applications in the current project.
- **Run Application**: Saves and executes the current application.
- **Export Applications**: Allows the export of one or more project applications.

- **Import Applications**: Imports a backup file that contains an application from another or the same project.
- **Source Code**: Allows you to visualize the generated source code of the current application.
- **Data in Session**: It displays all Scriptcase session variables, as well as those of the executed applications.

Tools

- **Data Dictionary**: Allows the user to model and standardize the tables fields (names, translation, field types, and general settings).
- **Express Edit**: Project applications mass editing.
- **HelpCase**: Enables the creation of project documentation.
- **External Libraries**: It allows you to import third-party libraries from the web (javascript, CSS, PHP, for example) or to create your libraries to use within your current project or entire Scriptcase projects.
- **Internal Libraries**: Allows you to create PHP scripts, similar to external libraries, but exclusively for PHP.
- **To-Do List**: Enables you to create a task list.
- **Messages**: Send messages among developers of the same Scriptcase installation.
- **Converters**:
    - **Version 6**: Converts your projects from ScriptCase version 6 to this current version.
    - **Version 7**: Converts your projects from ScriptCase version 7 to this current version.
    - **Version 8**: Converts your projects from ScriptCase version 8 to this current version.

Layout

- **CSS Applications (Themes)**: This option allows you to create or edit themes for the applications.
- **CSS Buttons**: This option allows you to create or edit button themes.
- **HTML Templates**: This option allows you to create or edit several HTML elements used within the applications.
- **CSS Menus**: This option allows you to edit the CSS of the menus.
- **Menu Icons**: Allows you to edit or create the icons of the menu application.
- **Image Manager**: This option manages the existing images in the project, also allowing you to add, delete and copy images.
- **Chart Themes**: This option allows you to edit or create the themes used within the Chart applications.
- **HTML Editor Templates**: Allows you to configure the functionalities from the Form and Control HTML editor field.

Local

- **Application Language**: Allows to define custom messages in different languages, according to the ones set during the project creation and the data dictionary.
- **Regional Settings**: Allows you to define some parameters of monetary units, date and numbers according to the Country or Region.

Modules

- **Security**: This option allows you to implement access rules to your system, creating a complete system with access control in your project.
- **Log**:
    - **Create/Edit Log Module**: Allows you to create or edit a log module.
    - **Applications Related to the log**: Allows you to reference which applications will use the log.
    - **Create Query with Log Report**: Allows you to create a Grid application that will display a log report.

Options

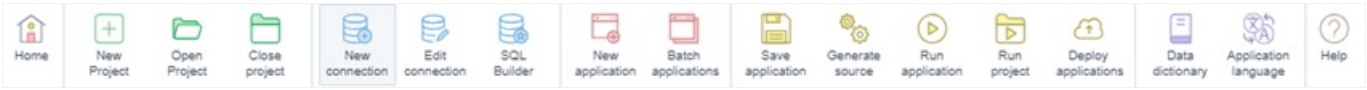
- **Settings**: Access all the ScriptCase settings.
- **My ScriptCase**: Allows you to customize the settings for the current user.
- **Change Password**: Allows the current user to change their password.
- **My Toolbar**: Allows you to change the position of the icons on the toolbar.

Help

- **WebHelp**: Allows you to access the manual.
- **Technical Support**: Access to ScriptCase online support.
- **Diagnosis**: Displays the settings of the Environment/Server where ScriptCase is installed.
- **Check Version**: Verifies if the version is updated.
- **Update Version**: Allows you to update ScriptCase to the latest version available.
- **License Registration**: Access the page to register your ScriptCase.
- **About**: Information about ScriptCase.

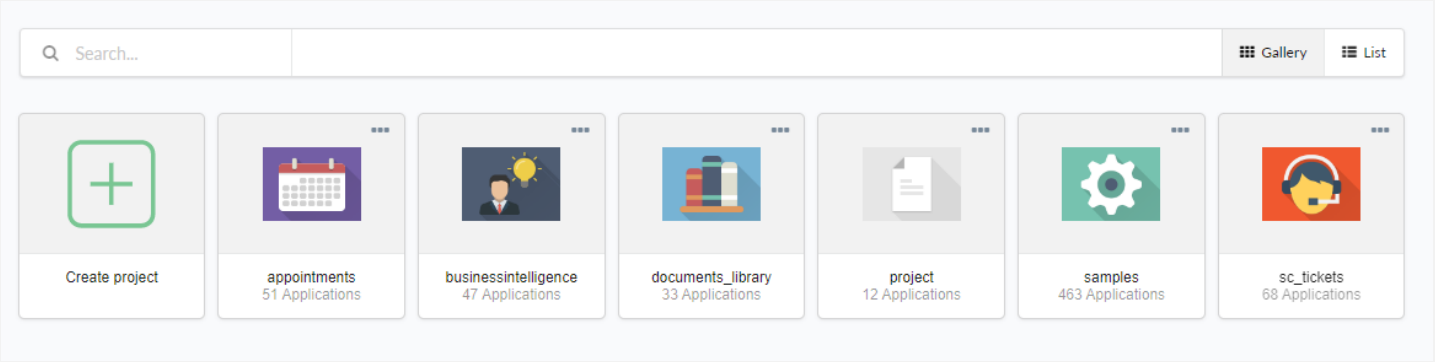
Scriptcase Toolbar

The toolbar is displayed right below the main menu, where you can find the most frequent used functions used. The toolbar is customizable, and you can create new features as well as edit the current ones. Learn how to customize the toolbar by clicking here.



- **Home**: Returns to the Application management interface.
- **New Project**: Starts a new project creation process.
- **Open Project**: Returns to project dash interface where you can see all the projects that your current user has access.
- **Close Project**: Closes the current project.
- **New Connection**: Displays a wizard to add new connections to the current project.
- **Edit Connection**: Edits the project database connections.
- **SQL Builder**: SQL editor to create or test queries inside the connected database. You can also run SQL commands using this feature.
- **New Application**: It opens the new application wizard for you to start a new application.
- **Batch Applications**: Creates various applications (Forms and Grids) simultaneously.
- **Save Application**: Saves the current application changes.
- **Generate Source Code**: Generate the Source Code for the existing Applications.
- **Run Application**: Saves the present application, generates the source code, and runs the applications.
- **Run Project**: Runs the complete project. To do this, you must enter the property in the project properties.
- **Deploy Applications**: Opens the wizard for the deployment procedure.
- **Data Dictionary**: Allows the user to define standards for the tables (Names, Field Types, etc.).
- **Application Language**: Feature to assist within translation of the sistem tables and internal messagers.
- **Help**: Scritcase's documentation.

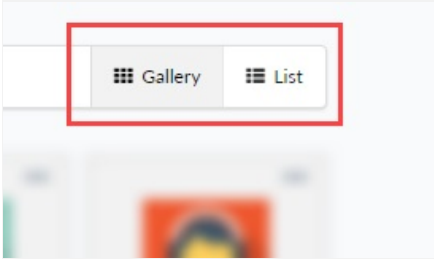
Project Dashboard



This is the initial Scriptcase screen, showed after login. It show the created projects that the user has access, these projects will be show as **Gallery**(default) format.

The definition of projects access should be made by the administrator user at the menu Options > Settings

Display Options



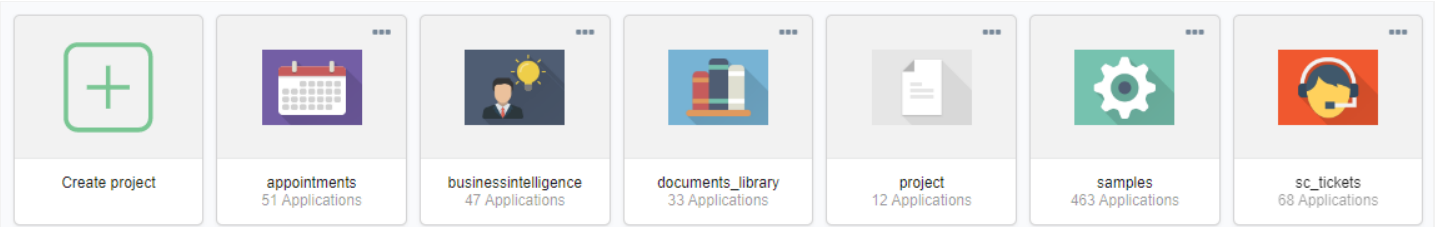
Through these bottom is possible to choose the project display between **Gallery**(Default) and **List**.

Gallery

The default display in the first time in Scriptcase, show the projects in cards with some basic information.

- Project image, it can be change in the project porperties.
- Project's name, choosed at create project.
- How many application at this version.

Sample of Project gallery



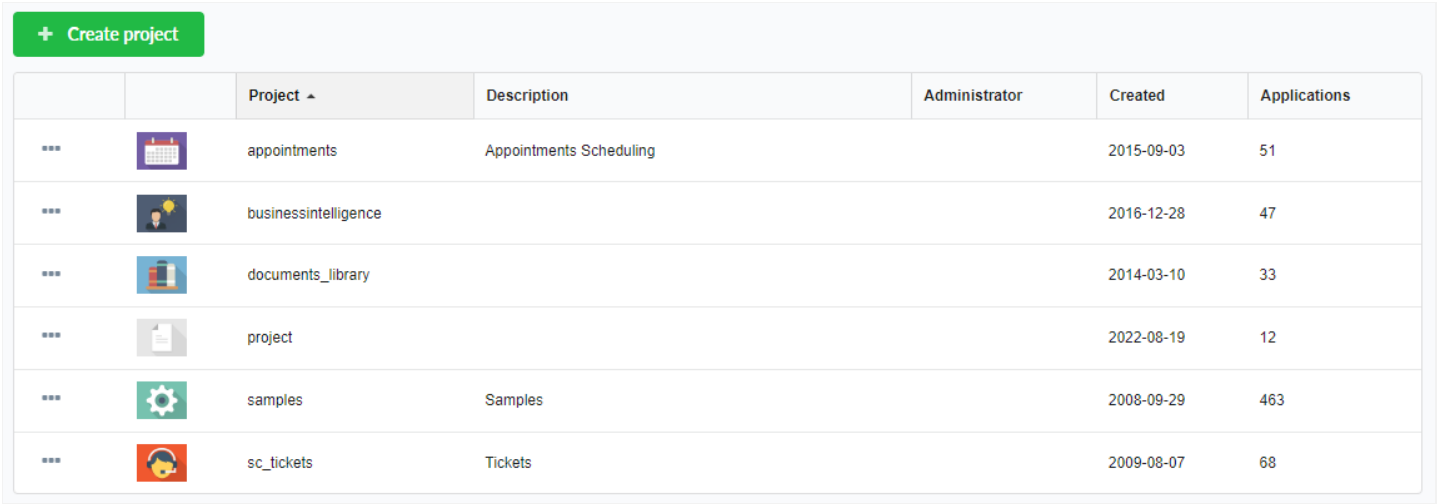
List

Show the projects in list with some more details about the project.

- Project image, it can be change in the project porperties.

- **Project** - Project's name, choosed at create project.
- **Description** - An description defined na at project creation. These description can be change at project porperties
- **Administrator** - List of users with administrator permission at the project. It is done at users manager in Options > Settings
- **Created** - Date of project creation.
- **Application** - How many application at this version.

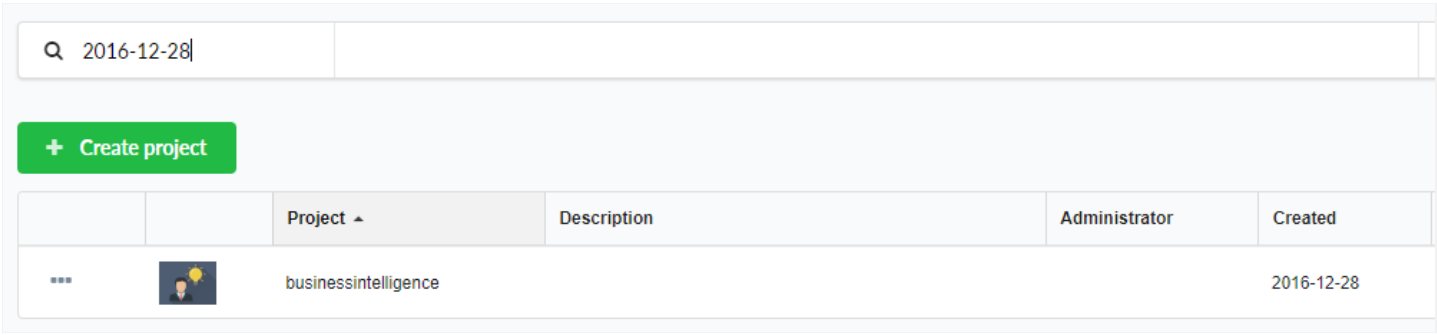
Sample of Project list



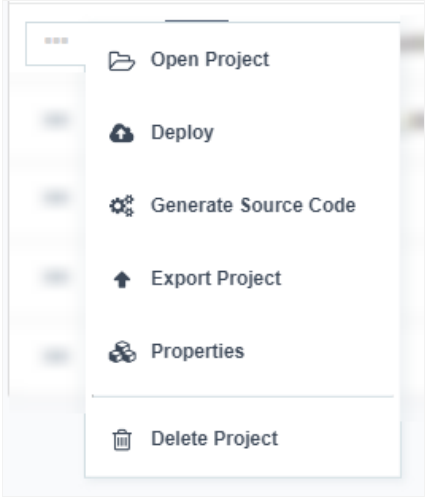
Search field

These field allow to search in the list of projects by the showed information, these way of display can be changed between **Gallery** or **List** also change how the field work because change the displayed information.

Sample of date search



Floating menu



These menu, can be access by the bottom ..., with some of project's options helping the developer access.

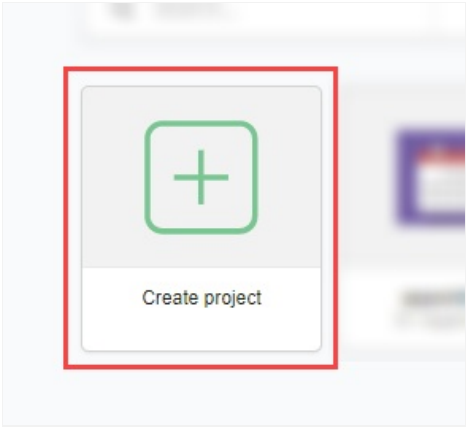
- Open project - Open the project at the home screen, showing the project's aplications.
- Deply - Open the project at deply screen.
- Generate source - Open the project and generate the source code of all aplications.
- Export Project - Open the project and start the export process.
- Properties - Open the project at the project's properties.
- Delete - Delete the project without accessing it.

Create project

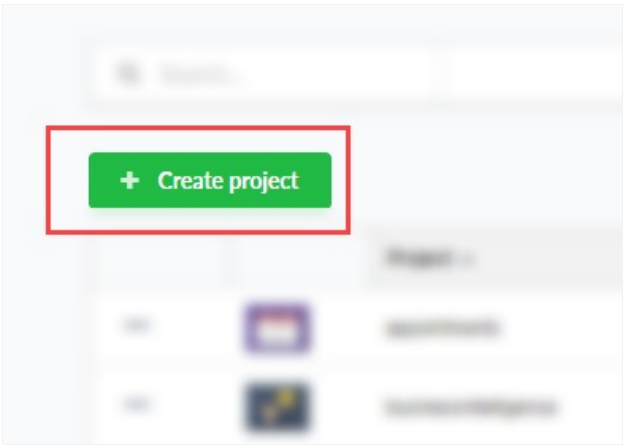
The change of project's display between **Gallery** and **List** also change the bottom display **Create project**.

Click here and see how to create a blank project or import a sample project

Create project sample showing at gallery

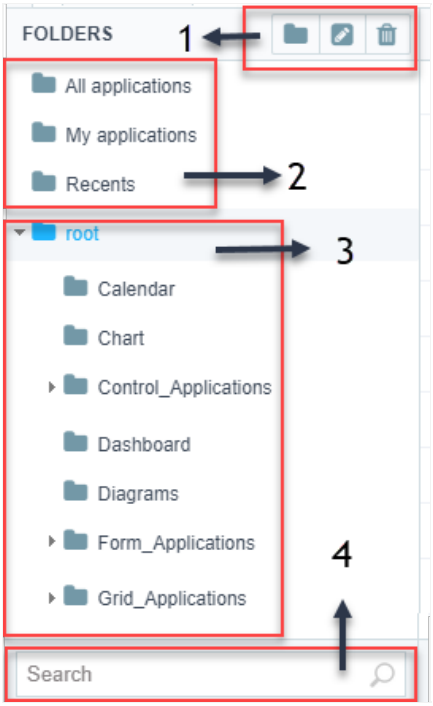


Create project sample showing at list



Project explorer

In this area, you can find all the folders created to organize the applications.



1 - Folders functions

Options for you to manage the folders (create, rename delete) to organize your application folders.

- **Create**: Creates a new folder.
- **Rename**: Renames the selected folder.
- **Delete**: Deletes the selected folder.

2 - Filters

Using the filter, you can quickly find an application per date or user. You can also display all the applications.

- **All Applications**: Lists all applications from the folders and the existing subfolders.
- **My Applications**: Lists all the applications from the folders and subfolders that the current user has created.
- **Recent**: List of all the new generated applications.

3 - Folders' structure

Scriptcase lists all application folders and sub-folders. It's recommendable for you to create folders to organize your project for future updates.

4 - Application search

Searches for the applications names or description.

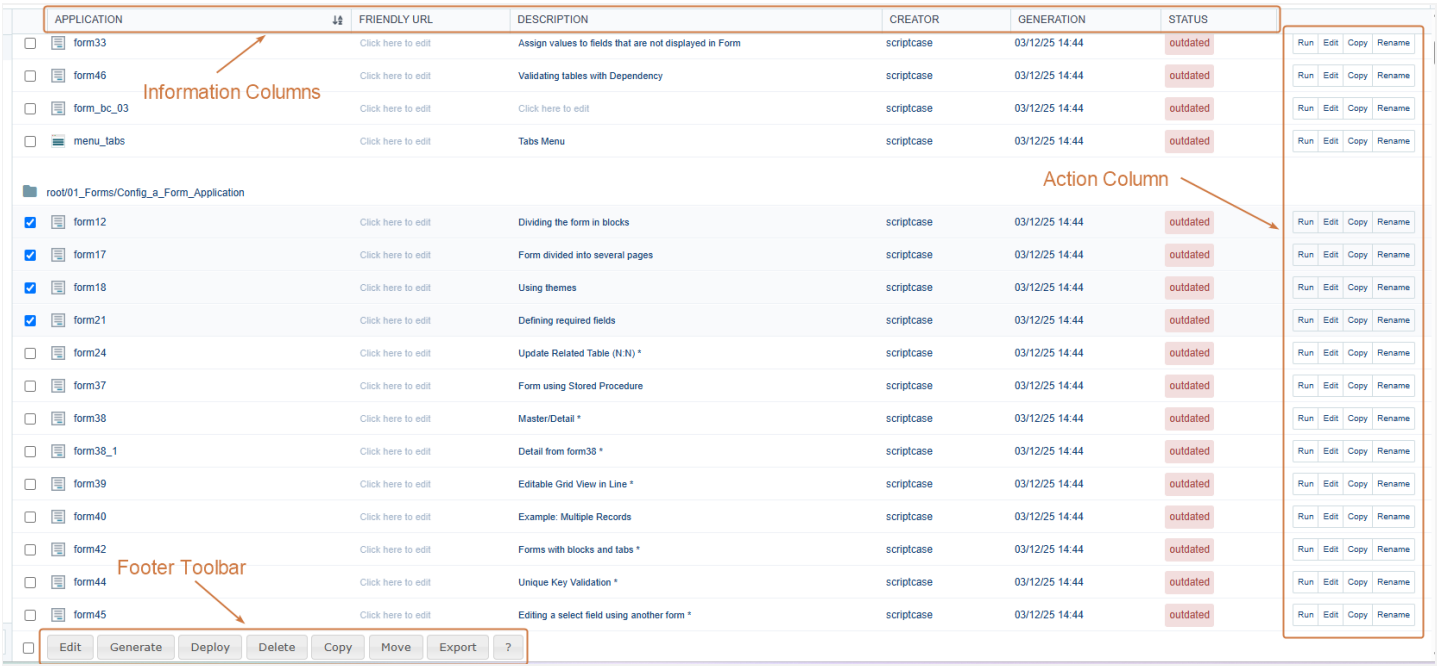
- **Search**: It searches by name or descriptions of the project. The search occurs within the selected folder, and the search results are highlighted in yellow.

List of Applications

The Application List is the display area for the contents of the folder selected in the *Project Explorer*. It allows you to view application details and perform quick changes, such as setting a Friendly URL or moving the application between project folders.

Additionally, you can perform bulk actions on selected applications using the button bar at the bottom.

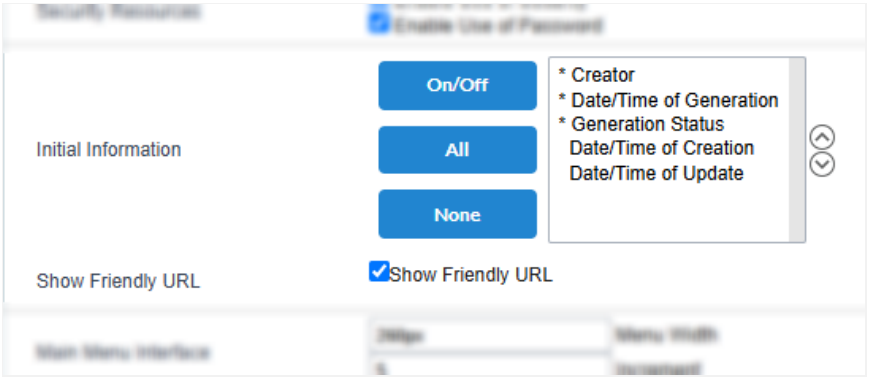
The *Application*, *Friendly URL*, and *Description* columns cannot be reordered or hidden.



Information Columns

You can customize the columns displayed in the Application List by adding or removing information according to your project needs.

To do this, access the **Settings > My Scriptcase** menu, under the *Initial Information* field.



Application

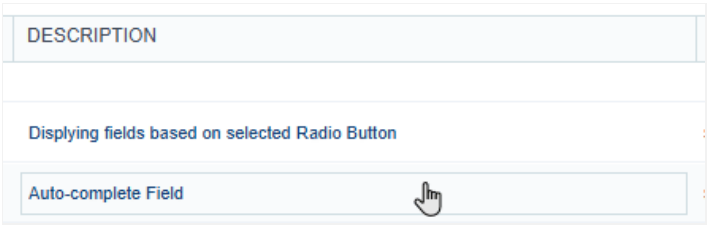
Column displaying the names of applications and folders within the folder selected in the *Project Explorer*.

Friendly URL

Allows you to quickly edit the application's Friendly URL without opening its configuration.

Description

Allows you to modify the application's description directly from the list without opening the settings screen.



Creator

Displays the login of the user who created the application.

Generation

Shows the date of the last source code generation. If blank, it means the application has not yet been compiled.

Status

Displays the application status in relation to the latest changes made in the development environment. The possible statuses are:

- **Updated**: The application has been successfully compiled, and the generated source code is in sync with the current configuration.
- **Outdated**: This status is used when:
  - Changes have been made but not yet generated;
  - Scriptcase has been updated (new version, generator adjustments, libraries, etc.).

Action Column

Exec Button

The *Exec* button allows you to run the application's code located in the `/app` folder.

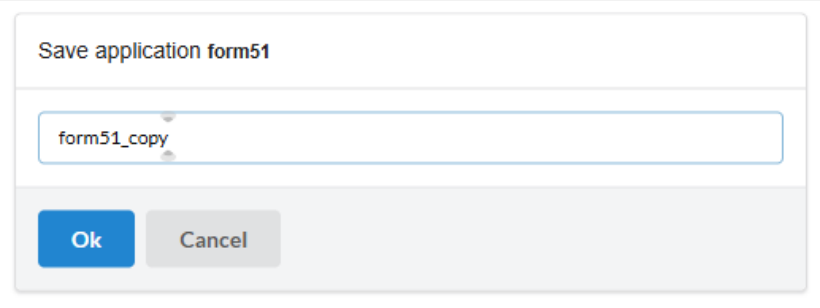
If the application's code has not yet been generated (compiled), the button will not function. In this case, it is necessary to generate the application's source code before using the *Exec* button in the Application List.

Edit Button

The *Edit* button opens the application's settings in the *taskbar*, allowing access to the application's editing environment.

Copy Button

The *Copy* button creates a duplicate of the application in the same project directory. When activated, a window will be displayed so the developer can define a new name for the copy.



Rename Button

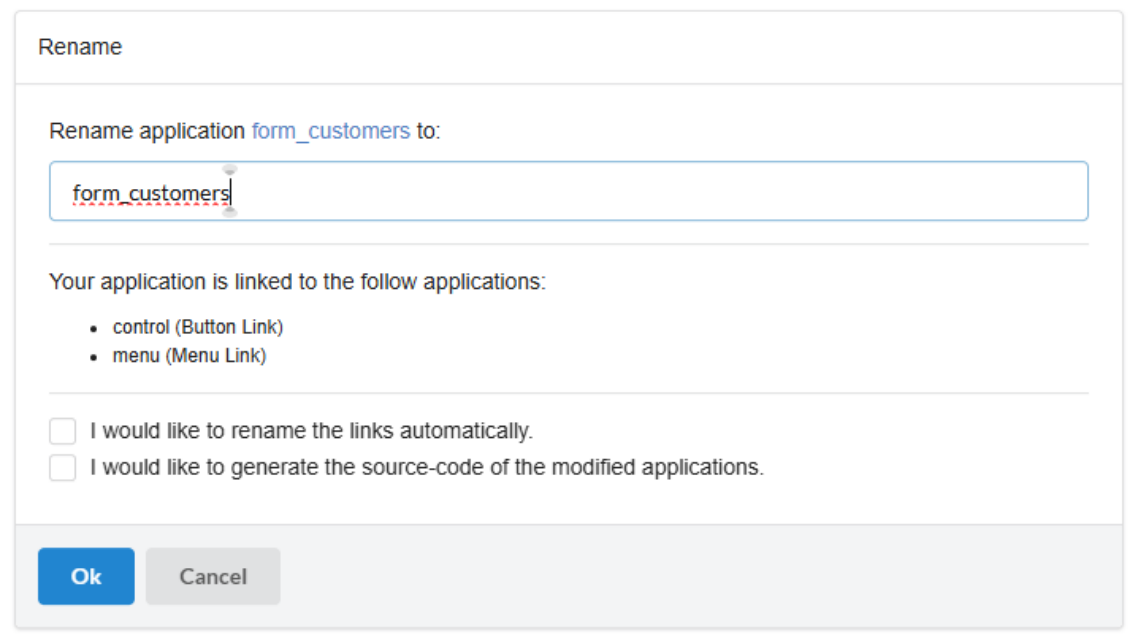
The *Rename* button allows you to change the application's name.

To perform this action, the application must not be open in the taskbar.

If the application is being referenced by other applications via linkage, those references will be listed below the name field. In these cases, two additional options will be available:

- **I want to rename linkage references automatically**: updates the application name in all linkages where it is referenced.
- **I want to generate the source code of the modified applications**: automatically recompiles all applications impacted by the renaming.

In the image below, for example, the application *app_form_sec_apps* is being renamed. It is referenced by the application *app_grid_sec_apps*.



Footer Toolbar

The footer buttons are only available when one or more applications in the list are selected.

Edit

Opens the settings of all selected applications for editing. Navigation will be directed to the last application opened, keeping the focus on it.

Generate

Generates the source code of the selected applications, updating the files in the `/app` folder. This action is essential after any development changes.

Deploy

Launches the publication wizard with all selected applications pre-checked, simplifying the export process to the production environment.

Delete

Permanently removes the selected applications from the project. A confirmation will be displayed before deletion.

Copy

Creates copies of the selected applications, allowing them to be duplicated into another project within Scriptcase.

Move

Moves the selected applications to a different folder within the same project, helping to better organize the structure in the Project Explorer.

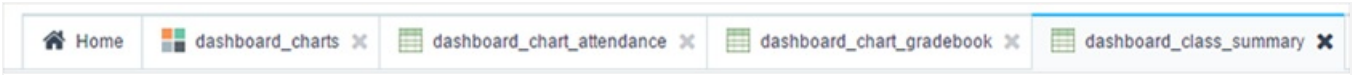
Export

Exports the selected applications in `.zip` format, enabling sharing or backup for later use in other Scriptcase environments.

Exported applications can only be imported into Scriptcase versions that are the same or newer.

Taskbar

This bar contains the tabs of all the applications that were opened for editing.

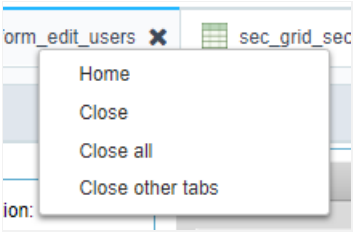


This bar is only displayed when at least one application has been opened.

The applications open in tabs in the taskbar where you can navigate among applications in a fast and easy way.

The "**x**" presented at the side of the Application name is used to close the application.

When you right-click on one of the tabs on the taskbar, you will see some tab closing options.



Keyboard Hotkeys

Keyboard hotkeys are keys or key combinations that provides an alternate way of doing something that you would normally do with a mouse.

The keyboard hotkeys are available in all browsers compatible with Scriptcase.

Browsers compatible with Scriptcase

Chrome	All versions
Firefox	Version 4 or above
Edge	All versions
Internet Explorer	Version 11 or above
Opera	Version 9 or above
Safari	Version 5.2 or above

Scriptcase Hotkeys

Efficiency is crucial to software development and the Keyboard Hotkeys are important to achieve that efficiently. You can see all the hotkeys that are now available in Scriptcase below:

Hotkeys	Actions
F1	Opens the documentation of the tool
CTRL+H or ⌘+H or F4	Displays the project homepage
CTRL+S or ⌘+S or F7	Saves the changes made in the application
CTRL+B or ⌘+B or F8	Generates the source code of the current application
CTRL+E or ⌘+E or F9	Runs the current application
F2	Opens the application creation screen
CTRL+I or ⌘+I or F6	Access to the SQL Builder
Alt+1~9	Accesses the selected tabs 1~9
Alt+Q	Closes the current tab
Alt+PgDown	Navigates back to the previous tab
Alt+PgUp	Navigates to the next tab

Events Editor Hotkeys - PHP

Using Sublime shortcut patterns, Scriptcase provides a series of shortcuts to streamline its development.

Hotkeys	Actions
Tab	Indents the line or selection
Shift+Tab	Removes line indentation or selection
Ctrl+Shift+K	Deletes the current line
Ctrl+Left	Moves the cursor to the next word
Ctrl+Right	Moves the cursor to the previous word
Ctrl+L	Selects the entire current line
Ctrl+Shift+L	Adds the cursor to all selected lines
Ctrl+	Cancels multiple cursors and moves the cursor to the first instance
Ctrl+Enter	Inserts a line after the current one
Ctrl+Shift+Enter	Inserts a line before the current one
Ctrl+Shift+D	Selects the next occurrence of selected text with the cursor
Ctrl+Shift+Space	Selects the current scope (within parentheses, braces, brackets ...)
Ctrl+Shift+M	Selects the broader context (keys, brackets, parentheses, page)
Ctrl+M	Moves the cursor to the closing or opening of the current context. (braces, brackets, parentheses, tags)
Ctrl+/	Comments or Uncomments line
Ctrl+J	Groups the line or selection
Ctrl+D	Duplicates line or selection
Ctrl+F9	Sorts the lines alphabetically, ignoring upper cases
Backspace	Intelligent Backspace (Removes indentation with line)
Ctrl+K Ctrl+K	Deletes text to the right
Ctrl+K Ctrl+U	Capital letters in the selection or cursor
Ctrl+K Ctrl+L	Lowercase letters in the selection or cursor
Ctrl+K Ctrl+C	Scrolls the line to the center of view
Ctrl+K Ctrl+Backspace	Deletes text to the left

Hotkeys	Actions
Ctrl+Shift+Up	Adds the cursor to the previous line
Ctrl+Shift+Down	Adds the cursor to the next line
Ctrl+F3	Finds the next occurrence of the selection or cursor
Ctrl+Shift+F3	Finds the next occurrence of the selection or cursor
Ctrl+Shift+[	Shows indented tag
Ctrl+Shift+]	Shows Inline tag
Ctrl+Shift+F	Activates search and replaces
Ctrl+F	Activates search
F3	Finds the next occurrence of the search
Shift+F3	Finds the previous occurrence of the search
Ctrl+Alt+I	Auto indentation of the selected code

Events Editor Hotkeys - HTML/CSS

Using Emmet's shortcut patterns, Scriptcase provides a series of shortcuts to streamline its development. With this plug-in it is possible to write HTML and CSS codes through CSS selectors.

Click Here to view the list of selectors.

For example, just write `ul> li.item $ {This is the item $} * 3` then press the "TAB" key and the following code will be generated.

```
 <li class="item1">This is item 1</li>
   <li class="item2">This is item 2</li>
   <li class="item3">This is item 3</li>
</ul>
```

To use the EMMET hotkeys in Scriptcase events, you must close and open the event PHP. Example: `?> ul> li.item $ {This is the item $} * 3 <? PHP`

Hotkeys	Actions
Tab	Enables EMMET abbreviations **Ex.** `ul>li.item${item $}*3`
Ctrl+Shift+A or ⌘+Shift+A	Envelopes the selected text in an EMMET abbreviation. Select the text you want to envelop, then use the keys, and enter the tag.
Ctrl+Alt+Right or ⌘+Alt+Right	Edit / selects the next node. Junction between tags, CSS blocks.
Ctrl+Alt+Left or ⌘+Alt+Left	Edit / selects the previous node. Junction between tags, CSS blocks.
Ctrl+Shift+Y or ⌘+Shift+Y	Validates and solves math expressions. Example: `2*8+4` .
Ctrl+Up or ⌘+Up	Increments numeric value `+1`
Ctrl+Down or ⌘+Down	Decrements numerical value `-1`
Ctrl+Shift+. or ⌘+Shift+.	Edit / selects the next item. Attribute, tag name
Ctrl+Shift+, or ⌘+Shift+,	Edit / selects the previous item. Attribute, tag name
Enter	Inserts indented line break.

Scriptcase Variables

Field Variables

In Scriptcase the fields are internally treated as local variables PHP, however, at Scriptcase interface it's necessary to inform these, using **keys**, so it's possible to retrieve or assign values to the application fields.

These fields can be used at Scriptcase events, but will not work from the **onScriptInit** event because only the events after that will have values in the fields.

In the examples below it's possible to visualize the operation of the fields.

Retrieving field value:

To assign the field value to a local variable to be used within an event, we can perform the steps as below:

$var_local = {application_field};

To assign the field value to a local variable to be used within an event, we can perform the steps as below:

[variavel_global] = {application_field};

Assigning Value to Field:

To assign a value to the field, we can do assignment in the same way as done with the variables, and if possible, according to the field type:

Field type Text: {field_text} = "This is a sample text";
Field type Number: {field_number} = 100;

Global Variables (Session Variables)

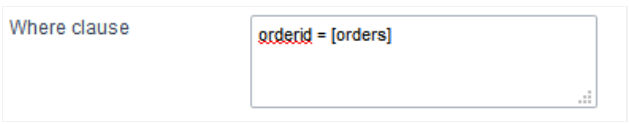
These are variables that are stored in the application session, these variables can be used to pass parameters through the applications in the project. These variables can be used at any application event.

Unlike the local and field variables, Global variables can be used in the SQL of the applications that allows the manual to be changed, and in the **WHERE** Clause of the Form application, thus enabling a dynamic use of the applications.

SQL Grid sample

```
SELECT
   orderid,
   customerid,
   employeeid,
   orderdate,
   requireddate,
   shippeddate,
   shipvia,
   freight,
   priceorder,
   shipcountry,
   shipregion,
   shipstate,
   shipcity,
   shipname,
   shipaddress,
   shippostalcode
FROM
   orders
WHERE orderid = [order]
```

Sample in the WHERE Clause of the Form



To define a global variable, you only have to enter it in between brackets, for example:

[global_variable].

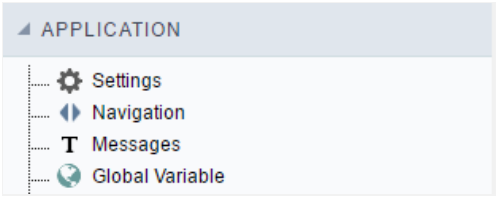
Assigning value to global variable:

[text_variable] = “This global has a text stored in it”;
[number_variable] = “This global has a number stored in it”;

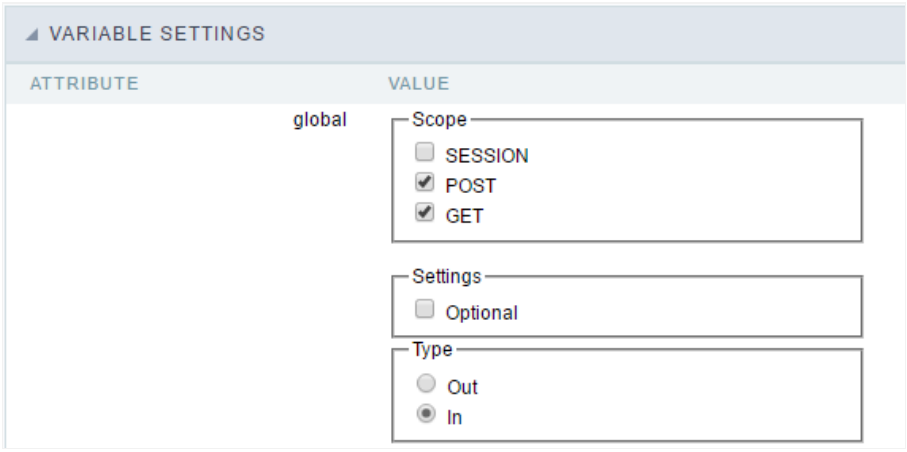
Passing values between applications

To pass values between applications, it is necessary to define the same variable in the applications that will use them, however, in the source application the variable must be defined as **Output** and in the destination application must be defined as **Input**.

To change the variable type, go to the menu **Application> Global Variables**.



After clicking on this menu will show the variables in the application and its settings.



In this configuration screen we have the following configuration options:

Attribute:

In Attributes we can see the variables in our application.

Value:

In Value we can see some options and configurations of our variable.

Type: This option defines whether the variable will be input or output.

Description:

Reports where the variable was declared by the developer.

Local Variables

Local variables are basically PHP's own variables. These variables must be set using a **$** dollar sign at the beginning and the variable names are case-sensitive.

These variables will only work in events or methods, using a variable in an event it will make it only available in that event.

Variable names follow the same rules as other labels in PHP. A valid variable name begins with a letter or underscore, followed by any number of letters, numbers, or underscores.

Below we can see some examples of local variables:

$text = "This variable stores a Text";
$number = "This variable stores a number";

For more information, visit the php variables.

EMMET Hotkeys

Below we have a list of EMMET CSS selectors in our event editor.

Remember that as they are CSS Selectors, to work as expected it is necessary to use inside the tag `<style> @m </style>`

CSS selectors	To open
@i or **@import**	`@import url( );`
@m or **@media**	`@media screen { }`
@f	`@font-face {font-family: ; src:url(); }`
@f+	`@font-face {font-family: 'FontName'; src: url('FileName.eot'); src: url('FileName.eot?#iefix') format('embedded-opentype'), url('FileName.woff') format('woff'), url('FileName.ttf') format('truetype'), url('FileName.svg#FontName') format('svg'); font-style: normal; font-weight: normal; }`
anim	`animation: ;`
anim-	`animation: name duration timing-function delay iteration-count direction fill-mode;`
animdel	`animation-delay: time;`
animdir or **animdir:n**	`animation-direction: normal;`
animdir:r	`animation-direction: reverse;`
animdir:a	`animation-direction: alternate;`
animdir:ar	`animation-direction: alternate-reverse;`
animdur	`animation-duration: 0s;`
animfm	`animation-fill-mode: both;`
animfm:f	`animation-fill-mode: forwards;`
animfm:b	`animation-fill-mode: backwards;`
animfm:bt or **animfm:bh**	`animation-fill-mode: both;`
animic	`animation-iteration-count: 1;`
animic:i	`animation-iteration-count: infinite;`
animn	`animation-name: none;`

CSS selectors	To open
animps	animation-play-state: running;
animps:p	animation-play-state: paused;
animps:r	animation-play-state: running;
animtf	animation-timing-function: linear;
animtf:e	animation-timing-function:ease;
animtf:ei	animation-timing-function:ease-in;
animtf:eo	animation-timing-function:ease-out;
animtf:eio	animation-timing-function:ease-in-out;
animtf:l	animation-timing-function:linear;
animtf:cb	animation-timing-function: cubic-bezier(0.1, 0.7, 0.1, 0.1);
ap	appearance: none;
!	!important"
pos or pos:r	position: relative;
pos:s	position:static;
pos:a	position:absolute;
pos:f	position:fixed;
t	top: ;
t:a	top:auto;
r	right: ;
r:a	right:auto;
b	bottom: ;
b:a	bottom:auto;
l	left: ;
l:a	left:auto;
z	z-index: ;
z:a	z-index:auto;

CSS selectors	To open
fl or **fl:l**	float: left;
fl:n	float:none;
fl:r	float:right;
cl or **cl:b**	clear: both;
cl:n	clear:none;
cl:l	clear:left;
cl:r	clear:right;
colm	columns: ;
colmc	column-count: ;
colmf	column-fill: ;
colmg	column-gap: ;
colmr	column-rule: ;
colmrc	column-rule-color: ;
colmrs	column-rule-style: ;
colmrw	column-rule-width: ;
colms	column-span: ;
colmw	column-width: ;
d or **d:b**	display: block;
d:n	display:none;
d:f	display:flex;
d:if	display:inline-flex;
d:i	display:inline;
d:ib	display:inline-block;
d:ib+	display: inline-block;\n*display: inline;\n*zoom: 1;
d:li	display:list-item;
d:ri	display:run-in;
d:cp	display:compact;

d:cp	display:compact;
d:tb	display:table;
d:itb	display:inline-table;
d:tbcp	display:table-caption;
d:tbcl	display:table-column;
d:tbclg	display:table-column-group;
d:tbhg	display:table-header-group;
d:tbfg	display:table-footer-group;
d:tbr	display:table-row;
d:tbrg	display:table-row-group;
d:tbc	display:table-cell;
d:rb	display:ruby;
d:rbb	display:ruby-base;
d:rbbg	display:ruby-base-group;
d:rbt	display:ruby-text;
d:rbtg	display:ruby-text-group;
v or **v:h**	visibility: hidden;
v:v	visibility:visible;
v:c	visibility:collapse;
ov or **ov:h**	overflow: hidden;
ov:v	overflow:visible;
ov:s	overflow:scroll;
ov:a	overflow:auto;
ovx or **ovx:h**	overflow-x: hidden;
ovx:v	overflow-x:visible;
ovx:s	overflow-x:scroll;
ovx:a	overflow-x:auto;

CSS selectors	To open
ovy	overflow-y: hidden;
ovy:v	overflow-y:visible;
ovy:h	overflow-y:hidden;
ovy:s	overflow-y:scroll;
ovy:a	overflow-y:auto;
ovs or ovs:s	overflow-style: scrollbar;
ovs:a	overflow-style:auto;
ovs:p	overflow-style:panner;
ovs:m	overflow-style:move;
ovs:mq	overflow-style:marquee;
zoo	zoom:1;
zm	zoom:1;
cp	clip: ;
cp:a	clip:auto;
cp:r	clip: rect(top right bottom left);
bxz	box-sizing: border-box;
bxz:cb	box-sizing:content-box;
bxz:bb	box-sizing:border-box;
bxsh	box-shadow: inset hoff voff blur color;
bxsh:r	box-shadow: inset hoff voff blur spread rgb(0, 0, 0);
bxsh:ra	box-shadow: inset h v blur spread rgba(0, 0, 0, .5);
bxsh:n	box-shadow:none;
m	margin: ;
m:a	margin:auto;
mt	margin-top: ;
mt:a	margin-top:auto;

CSS selectors	To open
mr	margin-right: ;
mr:a	margin-right:auto;
mb	margin-bottom: ;
mb:a	margin-bottom:auto;
ml	margin-left: ;
ml:a	margin-left:auto;
p	padding: ;
pt	padding-top: ;
pr	padding-right: ;
pb	padding-bottom: ;
pl	padding-left: ;
w	width: ;
w:a	width:auto;
h	height: ;
h:a	height:auto;
maw	max-width: ;
maw:n	max-width:none;
mah	max-height: ;
mah:n	max-height:none;
miw	min-width: ;
mih	min-height: ;
mar	max-resolution: res;
mir	min-resolution: res;
ori	orientation: ;
ori:l	orientation:landscape;
ori:p	orientation:portrait;
ol	outline: ;

CSS selectors	To open
ol:n	outline:none;
olo	outline-offset: ;
olw	outline-width: ;
olw:tn	outline-width:thin;
olw:m	outline-width:medium;
olw:tc	outline-width:thick;
ols	outline-style: ;
ols:n	outline-style:none;
ols:dt	outline-style:dotted;
ols:ds	outline-style:dashed;
ols:s	outline-style:solid;
ols:db	outline-style:double;
ols:g	outline-style:groove;
ols:r	outline-style:ridge;
ols:i	outline-style:inset;
ols:o	outline-style:outset;
olc	outline-color: #000;
olc:i	outline-color:invert;
bfv	backface-visibility: ;
bfv:h	backface-visibility:hidden;
bfv:v	backface-visibility:visible;
bd	border: ;
bd+	border: 1px solid #000;
bd:n	border:none;
bdbk or bdbk:c	border-break: close;
bdcl	border-collapse: ;

CSS selectors	To open
bdcl:c	border-collapse:collapse;
bdcl:s	border-collapse:separate;
bdc	border-color: #000;
bdc:t	border-color:transparent;
bdi	border-image: url();
bdi:n	border-image:none;
bdti	border-top-image:url();
bdti:n	border-top-image:none;
bdri	border-right-image:url();
bdri:n	border-right-image:none;
bdbi	border-bottom-image:url();
bdbi:n	border-bottom-image:none;
bdli	border-left-image:url();
bdli:n	border-left-image:none;
bdci	border-corner-image:url();
bdci:n	border-corner-image:none;
bdci:c	border-corner-image:continue;
bdtli	border-top-left-image:url();
bdtli:n	border-top-left-image:none;
bdtli:c	border-top-left-image:continue;
bdtri	border-top-right-image:url();
bdtri:n	border-top-right-image:none;
bdtri:c	border-top-right-image:continue;
bdbri	border-bottom-right-image:url();
bdbri:n	border-bottom-right-image:none;
bdbri:c	border-bottom-right-image:continue;
bdbli	border-bottom-left-image:url();

CSS	
bdbli	border-bottom-left-image:url();
bdbli:n	border-bottom-left-image:none;
bdbli:c	border-bottom-left-image:continue;
bdf or bdf:r	border-fit: repeat;
bdf:c	border-fit:clip;
bdf:sc	border-fit:scale;
bdf:st	border-fit:stretch;
bdf:ow	border-fit:overwrite;
bdf:of	border-fit:overflow;
bdf:sp	border-fit:space;
bdlen	border-length: ;
bdlen:a	border-length:auto;
bdsp	border-spacing:|;
bds	border-style: ;
bds:n	border-style:none;
bds:h	border-style:hidden;
bds:dt	border-style:dotted;
bds:ds	border-style:dashed;
bds:s	border-style:solid;
bds:db	border-style:double;
bds:dtds	border-style:dot-dash;
bds:dtdtds	border-style:dot-dot-dash;
bds:w	border-style:wave;
bds:g	border-style:groove;
bds:r	border-style:ridge;
bds:i	border-style:inset;
bds:o	border-style:outset;

CSS Editors To open

CSS selectors	To open
bdw	border-width: ;
bdtw	border-top-width: ;
bdrw	border-right-width: ;
bdbw	border-bottom-width: ;
bdlw	border-left-width: ;
bdt or **bt**	border-top: ;
bdt+	border-top: 1px solid #000;
bdt:n	border-top:none;
bdts	border-top-style: ;
bdts:n	border-top-style:none;
bdtc	border-top-color: #000;
bdtc:t	border-top-color:transparent;
bdr or **br**	border-right: ;
bdr+	border-right: 1px solid #000;
bdr:n	border-right:none;
bdrst	border-right-style: ;
bdrst:n	border-right-style:none;
bdrc	border-right-color: #000;
bdrc:t	border-right-color:transparent;
bdb or **bb**	border-bottom: ;
bdb+	border-bottom: 1px solid #000;
bdb:n	border-bottom:none;
bdbs	border-bottom-style: ;
bdbs:n	border-bottom-style:none;
bdbc	border-bottom-color: #000;
bdbc:t	border-bottom-color:transparent;

CSS selectors	To open
bdl or bl	border-left: ;
bdl+	border-left: 1px solid #000;
bdl:n	border-left:none;
bdls	border-left-style: ;
bdls:n	border-left-style:none;
bdlc	border-left-color: #000;
bdlc:t	border-left-color:transparent;
bdrs	border-radius: ;
bdtrrs	border-top-right-radius: ;
bdtlrs	border-top-left-radius: ;
bdbrrs	border-bottom-right-radius: ;
bdblrs	border-bottom-left-radius: ;
bg	background: #000;
bg+	background: #fff url() 0 0 no-repeat;
bg:n	background:none;
bg:ie	filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='x.png',sizingMethod='crop');
bgc	background-color: #fff;
bgc:t	background-color:transparent;
bgi	background-image:url();
bgi:n	background-image:none;
bgr	background-repeat: ;
bgr:n	background-repeat:no-repeat;
bgr:x	background-repeat:repeat-x;
bgr:y	background-repeat:repeat-y;
bgr:sp	background-repeat:space;
bgr:rd	background-repeat:round;
bga	background-attachment: ;

CSS selectors	To open
bga:f	background-attachment:fixed;
bga:s	background-attachment:scroll;
bgp	background-position: 0 0;
bgpx	background-position-x: ;
bgpy	background-position-y: ;
bgbk	background-break: ;
bgbk:bb	background-break:bounding-box;
bgbk:eb	background-break:each-box;
bgbk:c	background-break:continuous;
bgcp or bgcp:pb	background-clip: padding-box;
bgcp:bb	background-clip:border-box;
bgcp:cb	background-clip:content-box;
bgcp:nc	background-clip:no-clip;
bgo	background-origin: ;
bgo:pb	background-origin:padding-box;
bgo:bb	background-origin:border-box;
bgo:cb	background-origin:content-box;
bgsz	background-size: ;
bgsz:a	background-size:auto;
bgsz:ct	background-size:contain;
bgsz:cv	background-size:cover;
c	color: #000;
c:r	color: rgb(0, 0, 0);
c:ra	color: rgba(0, 0, 0, .5);
cm	/* */
cnt	content:' ';

CSS selectors	To open
cnt:n	content:normal;
cnt:oq	content:open-quote;
cnt:noq	content:no-open-quote;
cnt:cq	content:close-quote;
cnt:ncq	content:no-close-quote;
cnt:a	content:attr();
cnt:c	content:counter();
cnt:cs	content:counters();
tbl	table-layout: ;
tbl:a	table-layout:auto;
tbl:f	table-layout:fixed;
cps	caption-side: ;
cps:t	caption-side:top;
cps:b	caption-side:bottom;
ec	empty-cells: ;
ec:s	empty-cells:show;
ec:h	empty-cells:hide;
lis	list-style: ;
lis:n	list-style:none;
lisp	list-style-position: ;
lisp:i	list-style-position:inside;
lisp:o	list-style-position:outside;
list	list-style-type: ;
list:n	list-style-type:none;
list:d	list-style-type:disc;
list:c	list-style-type:circle;
list:s	list-style-type:square;

CSS selectors	To open
list:dc	list-style-type:decimal;
list:dclz	list-style-type:decimal-leading-zero;
list:lr	list-style-type:lower-roman;
list:ur	list-style-type:upper-roman;
lisi	list-style-image: ;
lisi:n	list-style-image:none;
q	quotes: ;
q:n	quotes:none;
q:ru	quotes: '00AB' '00BB' '201E' '201C';
q:en	quotes: '201C' '201D' '2018' '2019';
ct	content: ;
ct:n	content:normal;
ct:oq	content:open-quote;
ct:noq	content:no-open-quote;
ct:cq	content:close-quote;
ct:ncq	content:no-close-quote;
ct:a	content:attr( );
ct:c	content:counter( );
ct:cs	content:counters( );
coi	counter-increment: ;
cor	counter-reset: ;
va or va:t	vertical-align: top;
va:sup	vertical-align:super;
va:tt	vertical-align:text-top;
va:m	vertical-align:middle;
va:bl	vertical-align:baseline;
va:b	vertical-align:bottom;

CSS selectors	To open
va:tb	vertical-align:text-bottom;
va:sub	vertical-align:sub;
ta or ta:l	text-align: left;
ta:c	text-align:center;
ta:r	text-align:right;
ta:j	text-align:justify;
ta-lst	text-align-last: ;
tal:a	text-align-last:auto;
tal:l	text-align-last:left;
tal:c	text-align-last:center;
tal:r	text-align-last:right;
td or td:n	text-decoration: none;
td:u	text-decoration:underline;
td:o	text-decoration:overline;
td:l	text-decoration:line-through;
te	text-emphasis: ;
te:n	text-emphasis:none;
te:ac	text-emphasis:accent;
te:dt	text-emphasis:dot;
te:c	text-emphasis:circle;
te:ds	text-emphasis:disc;
te:b	text-emphasis:before;
te:a	text-emphasis:after;
th	text-height: ;
th:a	text-height:auto;
th:f	text-height:font-size;

CSS selectors	To open
th:t	text-height:text-size;
th:m	text-height:max-size;
ti	text-indent: ;
ti:-	text-indent:-9999px;
tj	text-justify: ;
tj:a	text-justify:auto;
tj:iw	text-justify:inter-word;
tj:ii	text-justify:inter-ideograph;
tj:ic	text-justify:inter-cluster;
tj:d	text-justify:distribute;
tj:k	text-justify:kashida;
tj:t	text-justify:tibetan;
tov or tov:e	text-overflow: ellipsis;
tov:c	text-overflow:clip;
to	text-outline: ;
to+	text-outline: 0 0 #000;
to:n	text-outline:none;
tr	text-replace: ;
tr:n	text-replace:none;
tt	text-transform: uppercase;
tt:n	text-transform:none;
tt:c	text-transform:capitalize;
tt:u	text-transform:uppercase;
tt:l	text-transform:lowercase;
tw	text-wrap: ;
tw:n	text-wrap:normal;
tw:no	text-wrap:none;

CSS selectors	To open
tw:n	text-wrap:none;
tw:u	text-wrap:unrestricted;
tw:s	text-wrap:suppress;
tsh	text-shadow: hoff voff blur #000;
tsh:r	text-shadow: h v blur rgb(0, 0, 0);
tsh:ra	text-shadow: h v blur rgba(0, 0, 0, .5);
tsh+	text-shadow: 0 0 0 #000;
tsh:n	text-shadow:none;
trf	transform: ;
trf:skx	transform: skewX(angle);
trf:sky	transform: skewY(angle);
trf:sc	transform: scale(x, y);
trf:scx	transform: scaleX(x);
trf:scy	transform: scaleY(y);
trf:scz	transform: scaleZ(z);
trf:sc3	transform: scale3d(x, y, z);
trf:r	transform: rotate(angle);
trf:rx	transform: rotateX(angle);
trf:ry	transform: rotateY(angle);
trf:rz	transform: rotateZ(angle);
trf:t	transform: translate(x, y);
trf:tx	transform: translateX(x);
trf:ty	transform: translateY(y);
trf:tz	transform: translateZ(z);
trf:t3	transform: translate3d(tx, ty, tz);
trfo	transform-origin: ;
trfs	transform-style: preserve-3d;

CSS selectors	To open
trs	transition: prop time;
trsde	transition-delay:time;
trsdu	transition-duration:time;
trsp	transition-property:prop;
trstf	transition-timing-function:tfunc;
lh	line-height: ;
whs	white-space: ;
whs:n	white-space: normal;
whs:p	white-space: pre;
whs:nw	white-space: nowrap;
whs:pw	white-space: pre-wrap;
whs:pl	white-space: pre-line;
whsc	white-space-collapse: ;
whsc:n	white-space-collapse:normal;
whsc:k	white-space-collapse:keep-all;
whsc:l	white-space-collapse:loose;
whsc:bs	white-space-collapse:break-strict;
whsc:ba	white-space-collapse:break-all;
wob	word-break: ;
wob:n	word-break:normal;
wob:k	word-break:keep-all;
wob:ba	word-break:break-all;
wos	word-spacing: ;
wow	word-wrap: ;
wow:nm	word-wrap:normal;
wow:n	word-wrap:none;
wow:u	word-wrap:unrestricted;

CSS selectors	To open
wow:s	word-wrap:suppress;
wow:b	word-wrap:break-word;
wm or **wm:lrt**	writing-mode:lr-tb;
wm:lrb	writing-mode:lr-bt;
wm:rlt	writing-mode:rl-tb;
wm:rlb	writing-mode:rl-bt;
wm:tbr	writing-mode:tb-rl;
wm:tbl	writing-mode:tb-lr;
wm:btl	writing-mode:bt-lr;
wm:btr	writing-mode:bt-rl;
lts	letter-spacing: ;
lts-n	letter-spacing:normal;
f	font: ;
f+	font: 1em Arial,sans-serif;
fw	font-weight: ;
fw:n	font-weight:normal;
fw:b	font-weight:bold;
fw:br	font-weight:bolder;
fw:lr	font-weight:lighter;
fs or **fs:i**	font-style:italic;
fs:n	font-style:normal;
fs:o	font-style:oblique;
fv	font-variant: ;
fv:n	font-variant:normal;
fv:sc	font-variant:small-caps;
fz	font-size: ;

CSS selectors	To open
fza	font-size-adjust: ;
fza:n	font-size-adjust:none;
ff	font-family: ;
ff:s	font-family:serif;
ff:ss	font-family:sans-serif;
ff:c	font-family:cursive;
ff:f	font-family:fantasy;
ff:m	font-family:monospace;
ff:a	font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;
ff:t	font-family: "Times New Roman", Times, Baskerville, Georgia, serif;
ff:v	font-family: Verdana, Geneva, sans-serif;
fef	font-effect: ;
fef:n	font-effect:none;
fef:eg	font-effect:engrave;
fef:eb	font-effect:emboss;
fef:o	font-effect:outline;
fem	font-emphasize: ;
femp	font-emphasize-position: ;
femp:b	font-emphasize-position:before;
femp:a	font-emphasize-position:after;
fems	font-emphasize-style: ;
fems:n	font-emphasize-style:none;
fems:ac	font-emphasize-style:accent;
fems:dt	font-emphasize-style:dot;
fems:c	font-emphasize-style:circle;
fems:ds	font-emphasize-style:disc;
fsm	font-smooth: ;

CSS selectors	To open
fsm:a	font-smooth:auto;
fsm:n	font-smooth:never;
fsm:aw	font-smooth:always;
fst	font-stretch: ;
fst:n	font-stretch:normal;
fst:uc	font-stretch:ultra-condensed;
fst:ec	font-stretch:extra-condensed;
fst:c	font-stretch:condensed;
fst:sc	font-stretch:semi-condensed;
fst:se	font-stretch:semi-expanded;
fst:e	font-stretch:expanded;
fst:ee	font-stretch:extra-expanded;
fst:ue	font-stretch:ultra-expanded;
op	opacity: ;
op+	opacity: ; filter: alpha(opacity=);
op:ie	filter:progid:DXImageTransform.Microsoft.Alpha(Opacity=100);
op:ms	-ms-filter:'progid:DXImageTransform.Microsoft.Alpha(Opacity=100)';
rsz	resize: ;
rsz:n	resize:none;
rsz:b	resize:both;
rsz:h	resize:horizontal;
rsz:v	resize:vertical;
cur	cursor:pointer;
cur:a	cursor:auto;
cur:d	cursor:default;
cur:c	cursor:crosshair;
cur:ha	cursor:hand;

CSS selectors	To open
cur:ha	cursor:hand;
cur:he	cursor:help;
cur:m	cursor:move;
cur:p	cursor:pointer;
cur:t	cursor:text;
fxd	flex-direction: ;
fxd:r	flex-direction:row;
fxd:rr	flex-direction:row-reverse;
fxd:c	flex-direction:column;
fxd:cr	flex-direction:column-reverse;
fxw	flex-wrap: ;
fxw:n	flex-wrap:nowrap;
fxw:w	flex-wrap:wrap;
fxw:wr	flex-wrap:wrap-reverse;
fxf	flex-flow: ;
jc	justify-content: ;
jc:fs	justify-content:flex-start;
jc:fe	justify-content:flex-end;
jc:c	justify-content:center;
jc:sb	justify-content:space-between;
jc:sa	justify-content:space-around;
ai	align-items: ;
ai:fs	align-items:flex-start;
ai:fe	align-items:flex-end;
ai:c	align-items:center;
ai:b	align-items:baseline;
ai:s	align-items:stretch;

CSS selectors	To open
ac	align-content: ;
ac:fs	align-content:flex-start;
ac:fe	align-content:flex-end;
ac:c	align-content:center;
ac:sb	align-content:space-between;
ac:sa	align-content:space-around;
ac:s	align-content:stretch;
ord	order: ;
fxg	flex-grow: ;
fxsh	flex-shrink: ;
fxb	flex-basis: ;
fx	flex: ;
as	align-self: ;
as:a	align-self:auto;
as:fs	align-self:flex-start;
as:fe	align-self:flex-end;
as:c	align-self:center;
as:b	align-self:baseline;
as:s	align-self:stretch;
pgbb	page-break-before: ;
pgbb:au	page-break-before:auto;
pgbb:al	page-break-before:always;
pgbb:l	page-break-before:left;
pgbb:r	page-break-before:right;
pgbi	page-break-inside: ;
pgbi:au	page-break-inside:auto;
pgbi:av	page-break-inside:avoid;

CSS selectors	To open
pgba	page-break-after: ;
pgba:au	page-break-after:auto;
pgba:al	page-break-after:always;
pgba:l	page-break-after:left;
pgba:r	page-break-after:right;
orp	orphans: ;
us	user-select:none;
wid	widows: ;
wfsm or wfsm:a	-webkit-font-smoothing:antialiased;
wfsm:s or wfsm:sa	-webkit-font-smoothing:subpixel-antialiased;
wfsm:n	-webkit-font-smoothing:none;

HTML Selectors	To open
!!!	<!DOCTYPE html>
!!!4t	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
!!!4s	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
!!!xt	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
!!!xs	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
!!!xxs	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
c	<!-- -->
cc:ie6	<!--[if lte IE 6]> <![endif]-->
cc:ie	<!--[if IE]> <![endif]-->
cc:noie	<!--[if !IE]><!--><!--<![endif]-->
!	Create an HTML5 structure

HTML Selectors	To open
a	`<a href=""></a>`
a:link	`<a href="http://"></a>`
a:mail	`<a href="mailto:"></a>`
abbr	`<abbr title=""></abbr>`
acr or **acronym**	`<acronym title=""></acronym>`
base	`<base href="">`
basefont	`<basefont>`
br	` `
frame	`<frame>`
hr	`<hr>`
bdo	`<bdo dir=""></bdo>`
bdo:r	`<bdo dir="rtl"></bdo>`
bdo:l	`<bdo dir="ltr"></bdo>`
col	`<col>`
link	`<link rel="stylesheet" href="">`
link:css	`<link rel="stylesheet" href="style.css">`
link:print	`<link rel="stylesheet" href="print.css" media="print">`
link:favicon	`<link rel="shortcut icon" type="image/x-icon" href="favicon.ico">`
link:touch	`<link rel="apple-touch-icon" href="favicon.png">`
link:rss	`<link rel="alternate" type="application/rss+xml" title="RSS" href="rss.xml">`
link:atom	`<link rel="alternate" type="application/atom+xml" title="Atom" href="atom.xml">`
link:im or **link:import**	`<link rel="import" href="component.html">`
meta	`<meta>`
meta:utf	`<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">`
meta:win	`<meta http-equiv="Content-Type" content="text/html;charset=windows-1251">`
meta:vp	`<meta name="viewport" content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">`

HTML Selectors	To open
meta:compat	<meta http-equiv="X-UA-Compatible" content="IE=7">
style	<style>
script	<script></script>
script:src	<script src=""></script>
img	<img src="" alt="">
img:s or **img:srcset**	<img srcset="" src="" alt="">
img:z or **img:sizes**	<img sizes="" srcset="" src="" alt="">
picture	<picture>
src or **source**	<source>
src:sc or **source:src**	<source src="" type="">
src:s or **source:srcset**	<source srcset="">
src:m or **source:media**	<source media="(min-width:)" srcset="">
src:t or **source:type**	<source srcset="" type="image/">
src:z or **source:sizes**	<source sizes="" srcset="">
src:mt or **source:media:type**	<source media="(min-width:)" srcset="" type="image/">
src:mz or **source:media:sizes**	<source media="(min-width)" sizes="" srcset="">
src:zt or **source:sizes:type**	<source sizes="" srcset="" type="image/">
iframe	<iframe src="" frameborder="0">
embed	<embed src="" type="">
object	<object data="" type="">
param	<param name="" value="">
map	<map name="">
area	<area shape="" coords="" href="" alt="">

area:d HTML Selectors	<area shape="default" href="" alt=""> To open
area:c	<area shape="circle" coords="" href="" alt="">
area:r	<area shape="rect" coords="" href="" alt="">
area:p	<area shape="poly" coords="" href="" alt="">
form	<form action="">
form:get	<form action="" method="get">
form:post	<form action="" method="post">
label	<label for="">
input	<input type="text">
inp	<input type="text" name="" id="">
input:h or input:hidden	<input type="hidden" name="">
input:t or input:text	<input type="text" name="" id="">
input:search	<input type="search" name="" id="">
input:email	<input type="email" name="" id="">
input:url	<input type="url" name="" id="">
input:p or input:password	<input type="password" name="" id="">
input:datetime	<input type="datetime" name="" id="">
input:date	<input type="date" name="" id="">
input:datetime-local	<input type="datetime-local" name="" id="">
input:month	<input type="month" name="" id="">
input:week	<input type="week" name="" id="">
input:time	<input type="time" name="" id="">
input:tel	<input type="tel" name="" id="">
input:number	<input type="number" name="" id="">
input:color	<input type="color" name="" id="">
input:c or	<input type="checkbox" name="" id="">

HTML Selectors	To open
input:c or **input:checkbox**	<input type="checkbox" name="" id="">
input:r or **input:radio**	<input type="radio" name="" id="">
input:range	<input type="range" name="" id="">
input:f or **input:file**	<input type="file" name="" id="">
input:s or **input:submit**	<input type="submit" value="">
input:i or **input:image**	<input type="image" src="" alt="">
input:b or **input:button**	<input type="button" value="">
isindex	<isindex>
input:reset	<input type="reset" value="">
select	<select name="" id=""></select>
select:d or **select:disabled**	<select name="" id="" disabled></select>
opt or **option**	<option value=""></option>
textarea	<textarea name="" id="" cols="30" rows="10"></textarea>
marquee	<marquee behavior="" direction=""></marquee>
menu:c or **menu:context**	<menu type="context"></menu>
menu:t or **menu:toolbar**	<menu type="toolbar"></menu>
video	<video src=""></video>
audio	<audio src=""></audio>
html:xml	<html xmlns="http://www.w3.org/1999/xhtml"></html>
keygen or **kg**	<keygen>
command or **cmd**	<command>
btn:s	<button type="submit"></button>
btn:r	<button type="reset"></button>

HTML Selectors	To open
btn:d	`<button disabled></button>`
fst:d	`<fieldset disabled></fieldset>`
bq	`<blockquote></blockquote>`
fig	`<figure></figure>`
figc	`<figcaption></figcaption>`
pic	`<picture></picture>`
ifr	`<iframe src="" frameborder="0"></iframe>`
emb	`<embed src="" type="">`
obj	`<object data="" type=""></object>`
cap	`<caption></caption>`
colg	`<colgroup></colgroup>`
fst or **fset**	`<fieldset></fieldset>`
btn	`<button></button>`
optg	`<optgroup></optgroup>`
tarea	`<textarea name="" id="" cols="30" rows="10"></textarea>`
leg	`<legend></legend>`
sect	`<section></section>`
art	`<article></article>`
hdr	`<header></header>`
ftr	`<footer></footer>`
adr	`<address></address>`
dlg	`<dialog></dialog>`
str	`<strong></strong>`
prog	`<progress></progress>`
mn	`<main></main>`
tem	`<template></template>`
datag	`<datagrid></datagrid>`

HTML Selectors	To open
datal	<datalist></datalist>
out	<output></output>
det	<details></details>
doc	Create an structure in HTML5 with the title "Documento"
doc4	Create an structure in HTML5 with the http-equiv="Content-Type" content="text/html;charset=UTF-8"
ri:d or **ri:dpr**	<img srcset="" src="" alt="">
ri:v or **ri:viewport**	<img sizes="" srcset="" src="" alt="">
ri:a or **ri:art**	Create an structure with the following tags '<picture>', '<source media="(min-width:)" srcset="">' e '<img src="" alt="">'
ri:t or **ri:type**	Create an structure with the following tags '<picture>', '<source srcset="" type="image/">' e '<img src="" alt="">'
html:4t	Create an structure in HTML4 using 'loose.dtd'
html:4s	Create an structure in HTML4 using 'strict.dtd'
html:xt	Create an structure in HTML4 using 'xhtml1-transitional.dtd
html:xs	Create an structure in HTML4 using 'xhtml1-strict.dtd'
html:xxs	Create an structure in HTML4 using 'xhtml11.dtd'
html:5	Create an structure in HTML5
ol+	Create an structure with the following tags ' ' e ' '
ul+	Create an structure with the following tags ' ' e ' '
dl+	Create an structure with the following tags '<dl>', '<dt>' e '<dd>'
map+	Create an structure with the following tags '<map name="">' e '<area shape="" coords="" href="" alt="">'
table+	CCreate an structure with the following tags '<table>', '<tr>' e '<td>'
colgroup+ or **colg+**	Create an structure with the following tags '<colgroup>' e '<col>'
tr+	Create an structure with the following tags '<tr>' e '<td>'
select+	Create an structure with the following tags '<select name="" id="">' e '<option value=""></option>'
optgroup+ or **optg+**	Create an structure with the following tags '<optgroup>' e '<option value=""></option>'

pic+ HTML Selectors	Create an structure with the following tags '<picture>', '<source srcset="">' e '<img src="" alt="">' To open

Scriptcase Projects Overview

The notion of a project in Scriptcase is fundamental for the organization and structured development of applications. Each project groups together all the resources needed to create complete systems, from registration forms and queries (grids) to reports and charts. Below is an overview of how projects work in Scriptcase and the main steps to create and manage one:

What is a Project in Scriptcase?

A project is a set of applications (forms, grids, dashboards, menus, etc.) that collectively make up your system. Within it, you control the database connection, layout settings, languages, libraries, and all the code necessary for the applications to run. This way, each project operates independently, allowing for clear organization and easier maintenance.

Creating a New Project

1. **Access Scriptcase**: After logging in, select the option to create a new project.
2. **Define Name and Description**: Give it a title that represents the goal of the system being developed. The description serves as a reference to understand the project's purpose.
3. **Language and Regional Settings**: Choose the main language and any localization configurations.
4. **Database Connection**: Select the type of database to be used (MySQL, PostgreSQL, SQL Server, Oracle, etc.) and define the connection parameters.
5. **Finish and Start Development**: Once the settings are finalized, you can begin creating the necessary applications (Forms, Grids, Reports, etc.) within the project.

Structure and Organization

Within a project, each application can be created and organized in folders, making it easier to access and maintain the code. In addition, Scriptcase allows for the sharing of internal libraries and the configuration of visual themes to ensure a consistent look and feel across all applications.

Best Practices

- **Backup and Versioning**: Perform periodic exports of the project using the "Export Project" feature to preserve settings and changes.
- **Security**: Use Scriptcase's built-in access control features to manage users and permissions.
- **Interface Standardization**: Utilize Default Values, layout tools, and internal libraries to maintain a uniform visual style across all applications.
- **Team Collaboration**: When multiple developers are involved in the project, establishing a version control workflow and clear communication is essential to avoid conflicts.

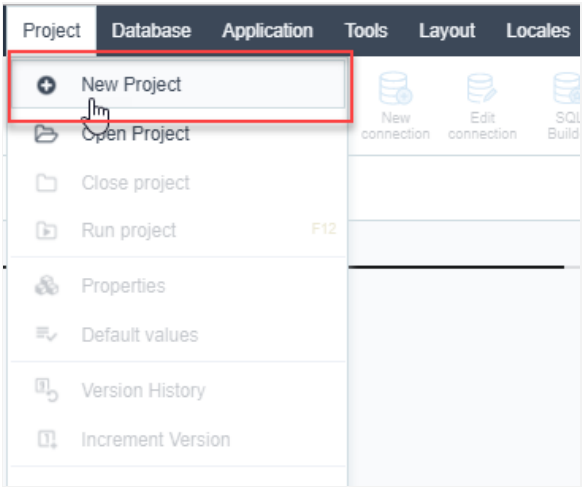
> **Related Videos** ▷

New Project

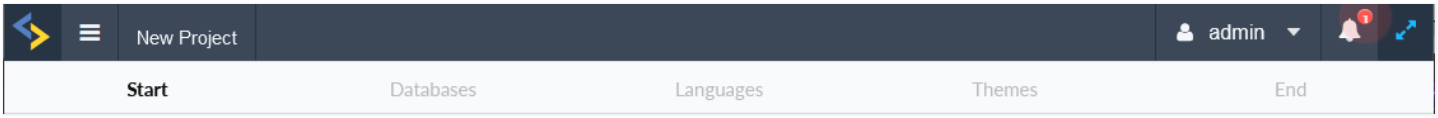
In this article you learn how to create a new project within Scriptcase.

A system made with Scriptcase has always to start by creating a project. A project for Scriptcase is a set of applications with some standard settings. During the project creation, you define some essential options as name and description, languages in which your project might be available, a layout template for the applications, and informs the database system connection data in which the applications fetch the information to form the system.

To start a new project you can access the menu **"Project > New Project"**.

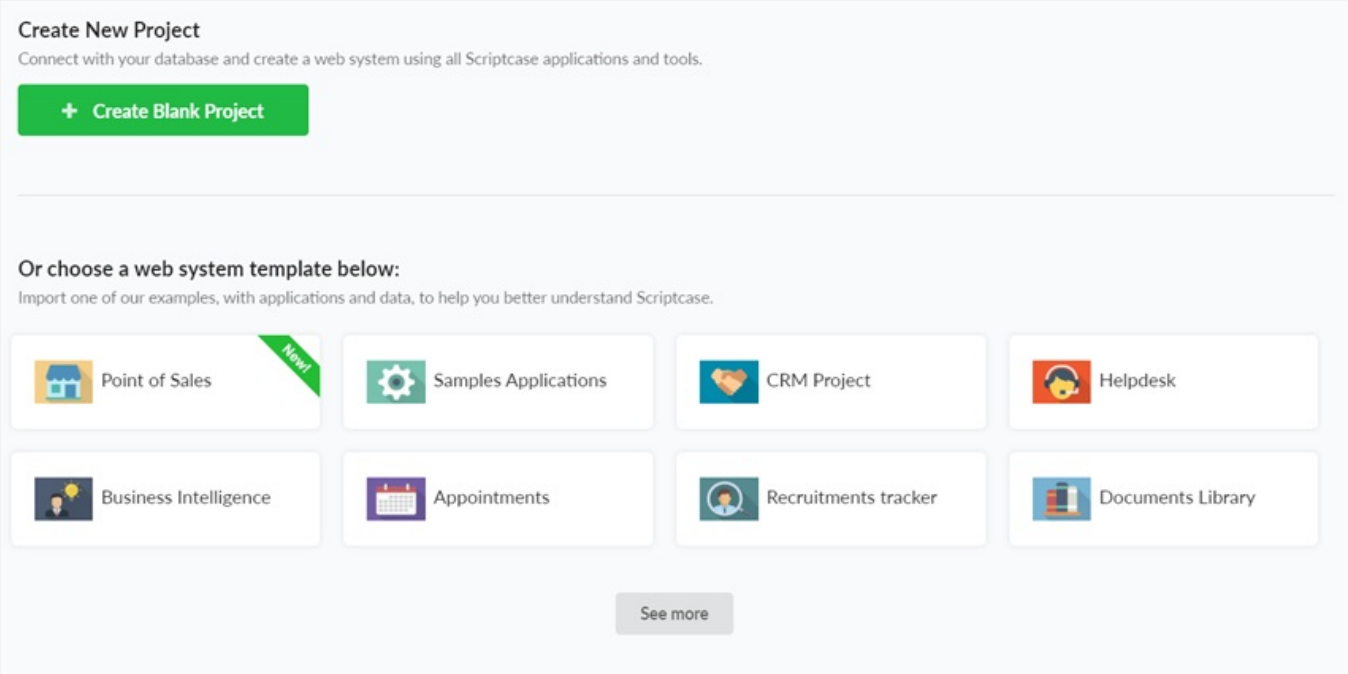


The project creation process has five steps, which are: **Start, Database, Languages, Themes and End** .



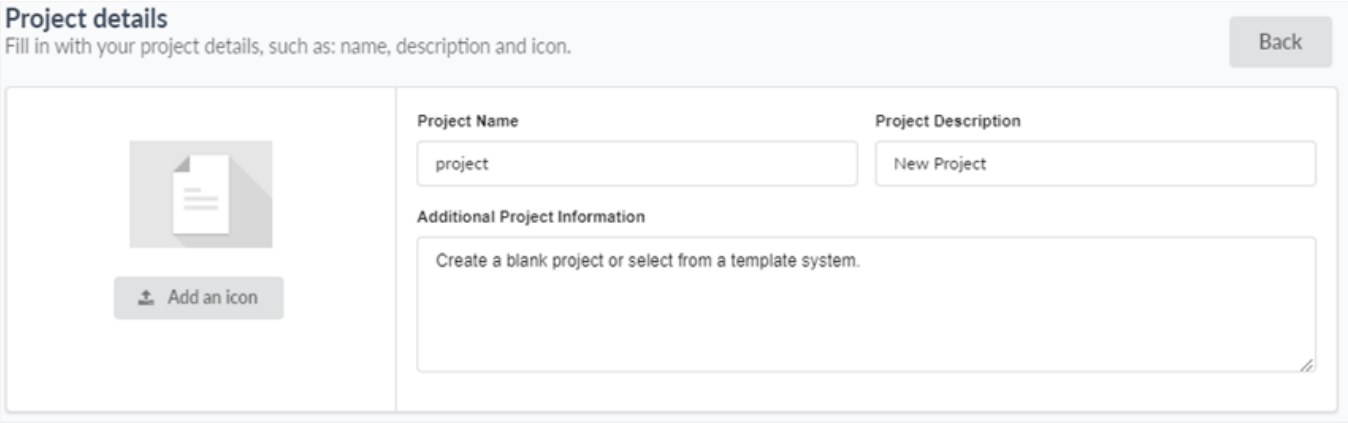
Start

In this step, we should choose between a **Blank project**, where we have to create our database connection and every application from scratch, or one select of our **Template projects**, that can serve as a template to the creation of your system.



NOTE: Click on “See more” button to have access to our Sample Projects and click in the link here to know how to import them.

After we select **Create Blank project**, it is mandatory to define a name to our project.



Project name

The only mandatory setting for this step is that the name must contain 1 to 32 alphanumeric characters and does not allow it to start with a number.

Project description

A not required that allows us to define a description for the project.

After the creation of the project, we can see the description in the dialog box when we place the mouse over the project when we use the gallery view or in the Description column when the project’s view is in **list** format.

Additional Project Information

Not required fields that allow us to add additional information to the project. This information is within the project properties.

Image

Not required fields that allow us to define an image to represent our project.

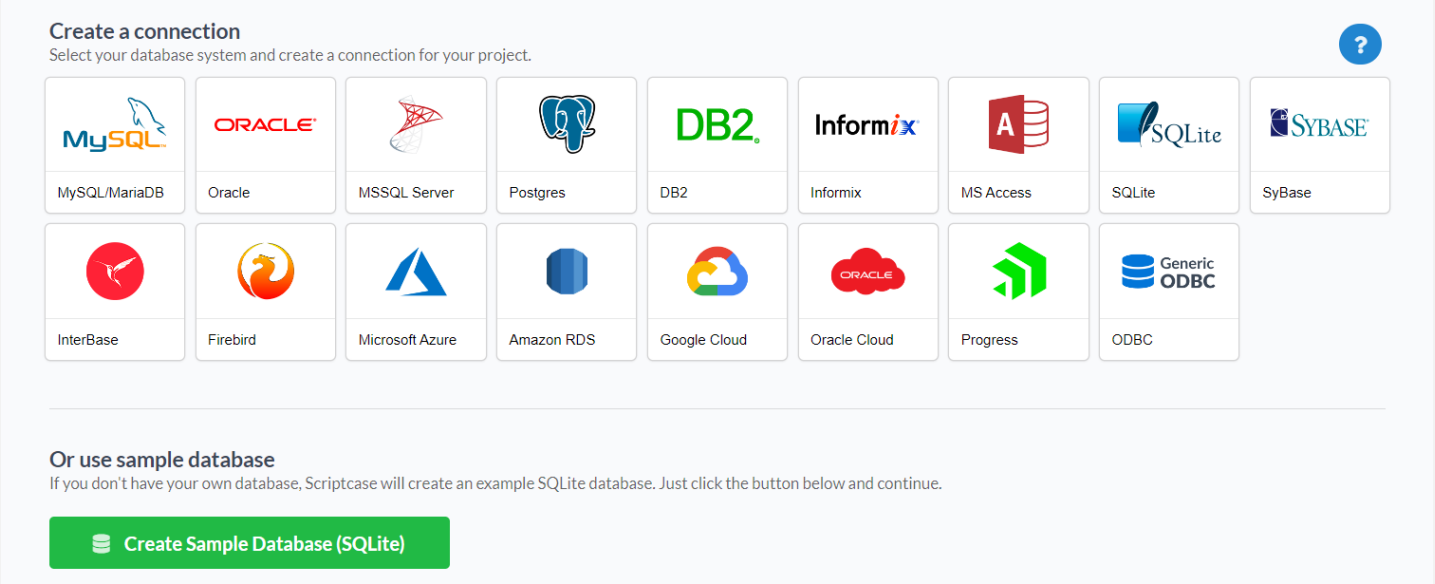
You can change all non-mandatory fields in this step, and other project settings after project creation by accessing the option Project Properties.

Databases

During the connection creation, Scriptcase lists all database systems according to your license. If you're evaluating, Scriptcase displays all database systems available. You have to select your database and proceed to the next step to inform the database connection details.

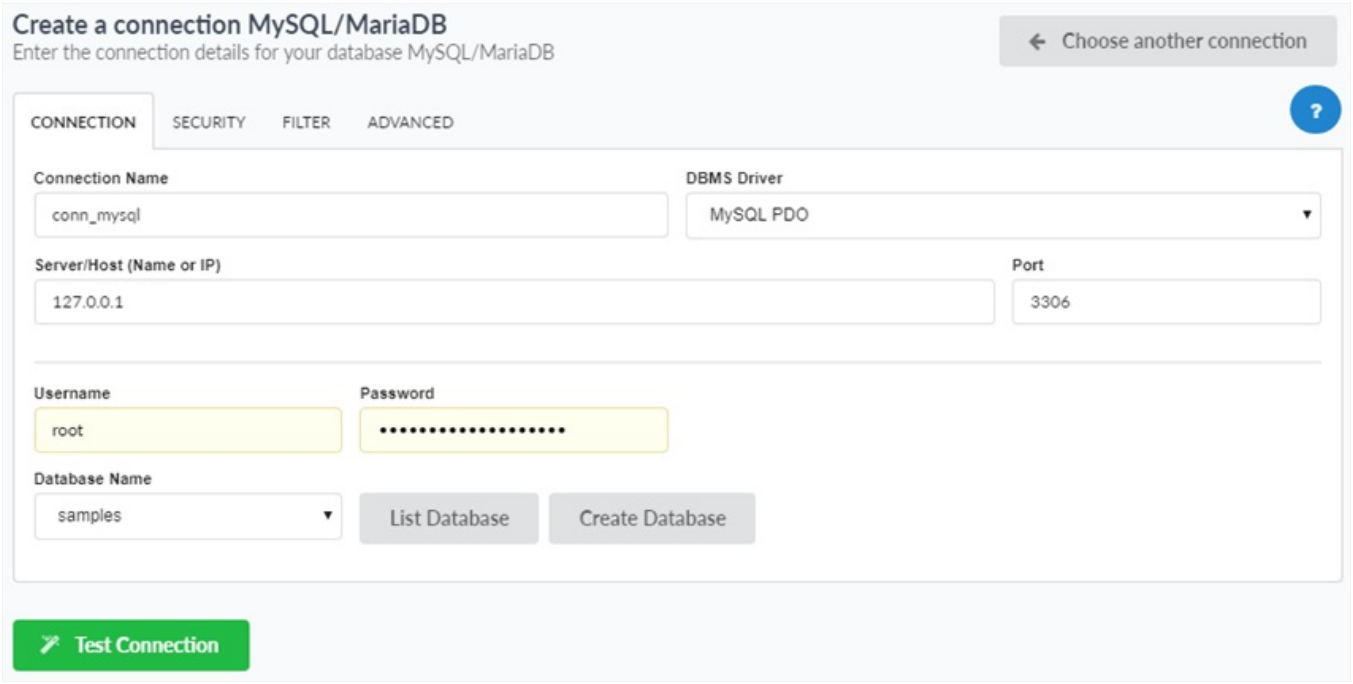
You need to have a database system installed and the permissions to connect before you proceed.

Note: If you don't have a database available to create the project, Scriptcase provides a sample database in SQLite with the tables ready for the user to proceed with the creation of the project. To create this database just click on the button **Create Sample Database (SQLite).**



The information required to connect is different for each database. In this example we will be using MySQL. To learn more about other connections please access this link.

Using Scriptcase you are able to create as well as to manage database projects and tables. During this step you will need to create or connect to, at least, one database project. We are going to use a "Samples" one. After the project creation you are able to use the Database manager and SQL Builder features to manage the database project and queries.



Database connection details.

- **Connection Name**: Define the name of the connection. You are not able to change this name in the future.
- **DBMS Driver** : Defined as MySQL PDO. It can also be MySQLi.
- **Server/Host (Name or IP)**: Enter the IP of the server where the database is. If the database is on the same machine of Scriptcase, you can use the IP **127.0.0.1** or **localhost**. If the Database is
- **Port**: Define the port of the connection. By default, the port is 3306.
- **User**: Inform the username that you use to connect to your database.
- **Password**: Inform the password that you use to connect to your database.
- **Database Name**: Click on the "List Database" and select the desired database.
- **Test Connection**: Displays the connection status if it succeeded or not.

All available connections are listed according to on your license type.Go to our on-line store and view the available licenses.

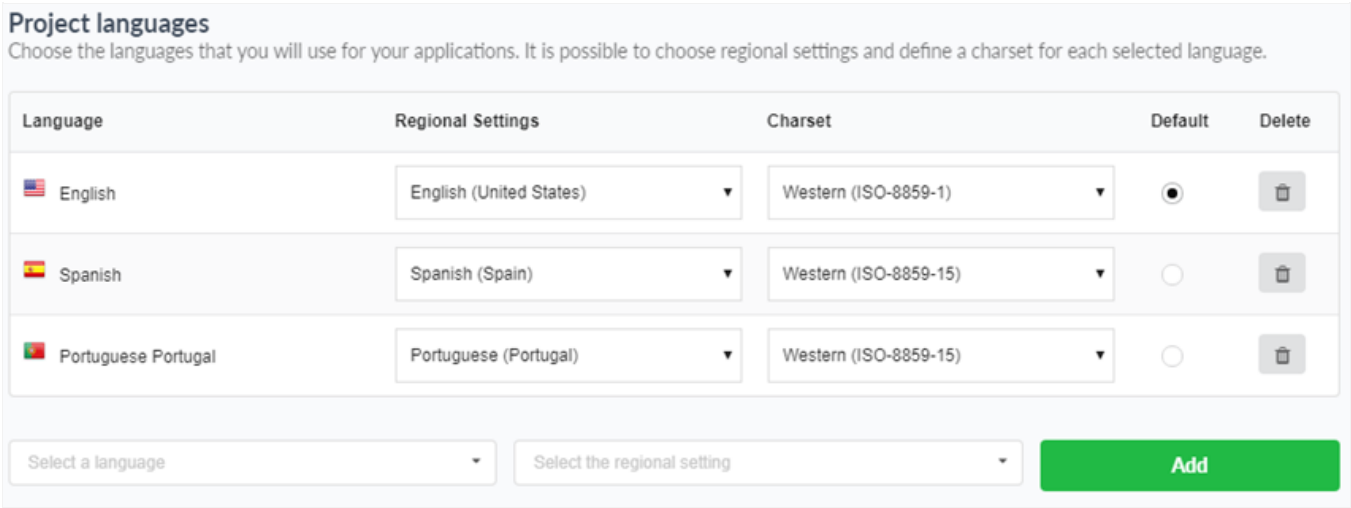
If you have connection problems, access our free support to get assistance by chat or remote access.

Language

In this step, you should select the languages that are available for system development.

Scriptcase can easily create multilingual systems. You need to add more than one language in this step of the project. All the languages you choose are available to the project, and you must choose one as standard. You can also have the same language more than once using different regional settings.

Scriptcase will automatically translate and apply the regional settings to your project applications (for buttons and internal messages for example). You can also use another tool from Scriptcase called Data dictionary to import and translate your database fields. You will be able to see, create or customize the messages inside the menu > Locales > Application language.



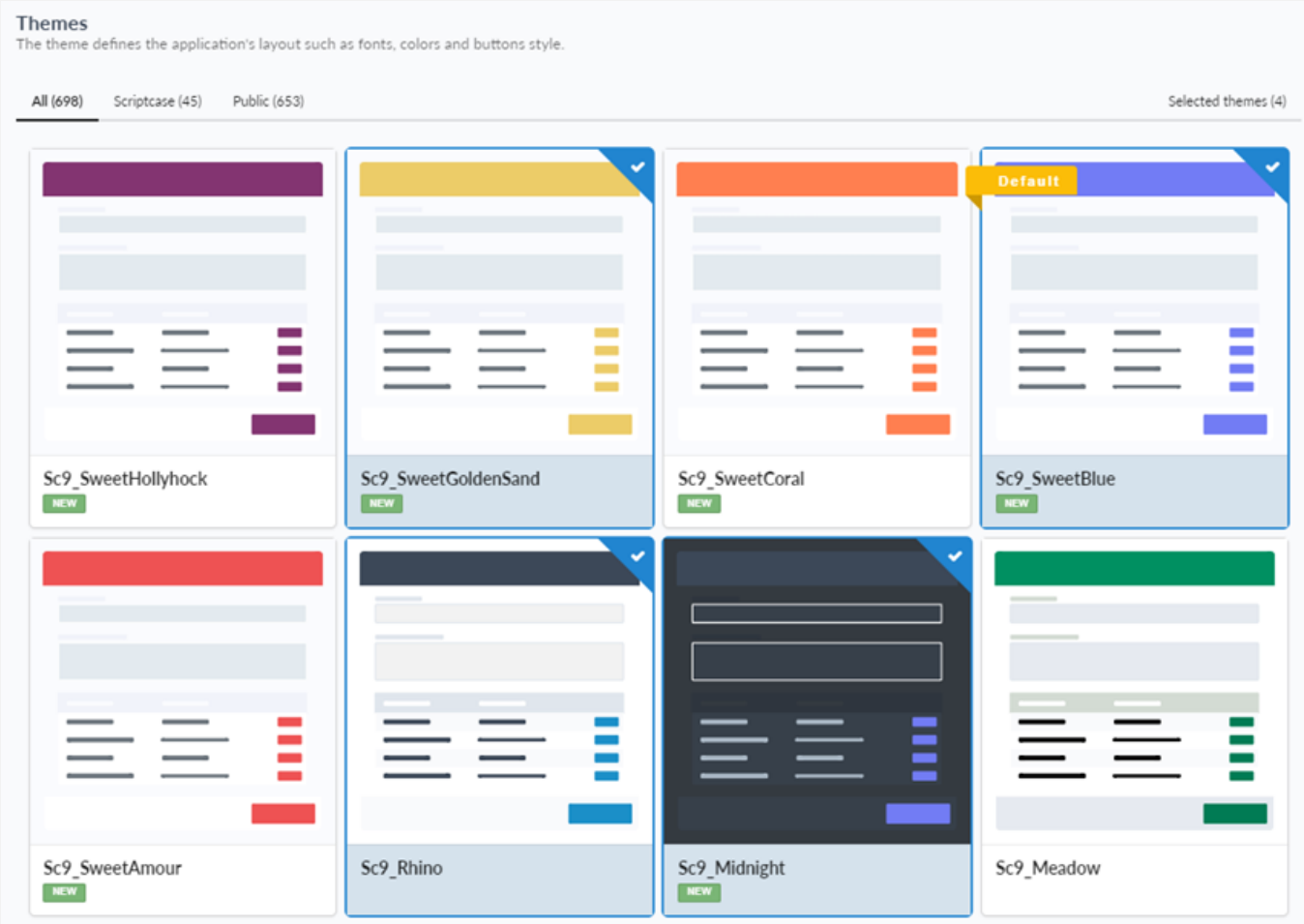
- **Language** - List of languages that are in the projects.
- **Regional Settings** - Contains the standard regional settings for the selected language. Regional Settings allows you to define some parameters of monetary units, dates, and numbers according to the country or region where your application.
- **Charset** - Sets the character encoding format for the application. This encoding must be the same used in your database. By default, the charset comes according to the language, and you can change after creation in project properties.
- **Default** - Sets the project default language. Project applications are performed and displayed using this default language; it can also be changed individually within the application settings.
- **Delete** - Removes the language of the project. In this way, the language removed will no longer be available. It may also be included or deleted after the project created within the Menu > Project > Properties.

After selecting the language, regional setting, and charset, click “Add” and then “Save” to move forward.

Themes

In the themes selection, you can choose one more than templates for your project. Within this template you will have a complete theme for all applications you create within the project. Ready to use screens with pre made CSS and HTML to all application types.

You can also customize or create new themes from the templates. Once you are inside the projects, access the menu Layout > CSS Application (Themes).



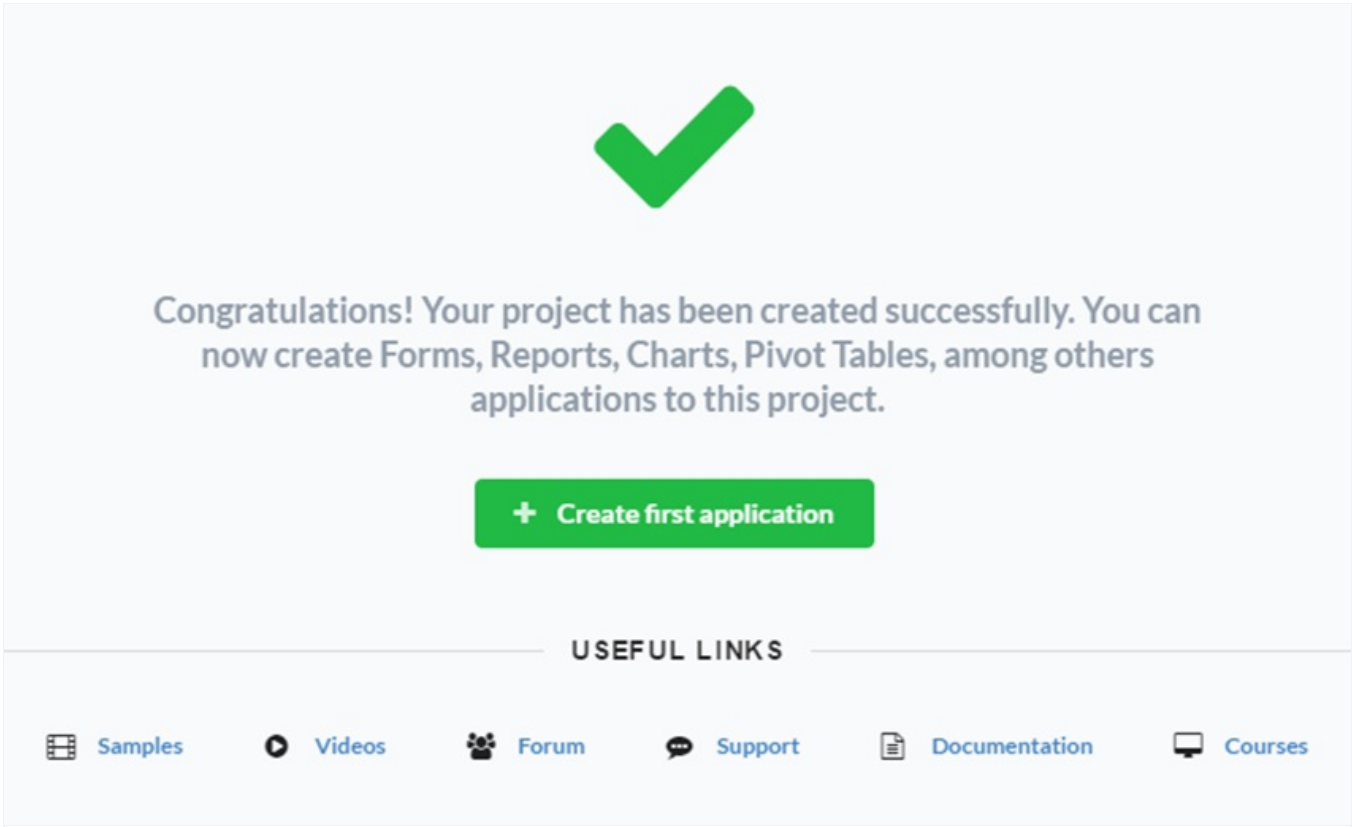
- **All** - Lists all the available themes for use in your project.
- **Scriptcase** - Lists all Scriptcase standards themes.
- **Public** - Lists all themes defined in the Public level.
- **Selected themes** - Lists the themes selected for use in the project.

To add a theme to your project, you only need to select it, and then it is going to be listed inside the selected themes. You should select a theme to use as default, for this click on the “default” icon of the selected theme. The default theme is the one selected every time you create a new application, and you can also select a different one according to the options available within the selected themes. You can also access this option in the future to add new themes.

When you add more than one theme, you can dynamically switch themes before creating the applications, by adding a toolbar button within the applications or using the macro sc_set_theme.

End

The project has been created successfully with the basic settings, and now you can create the applications. Visit the Applications menu to learn more about each application and to see the steps to create them.

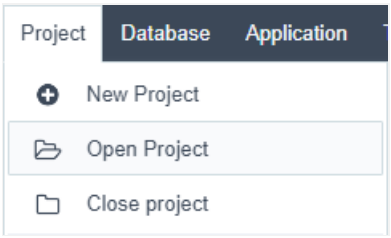


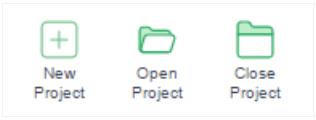
Related Videos ▷

Open Project

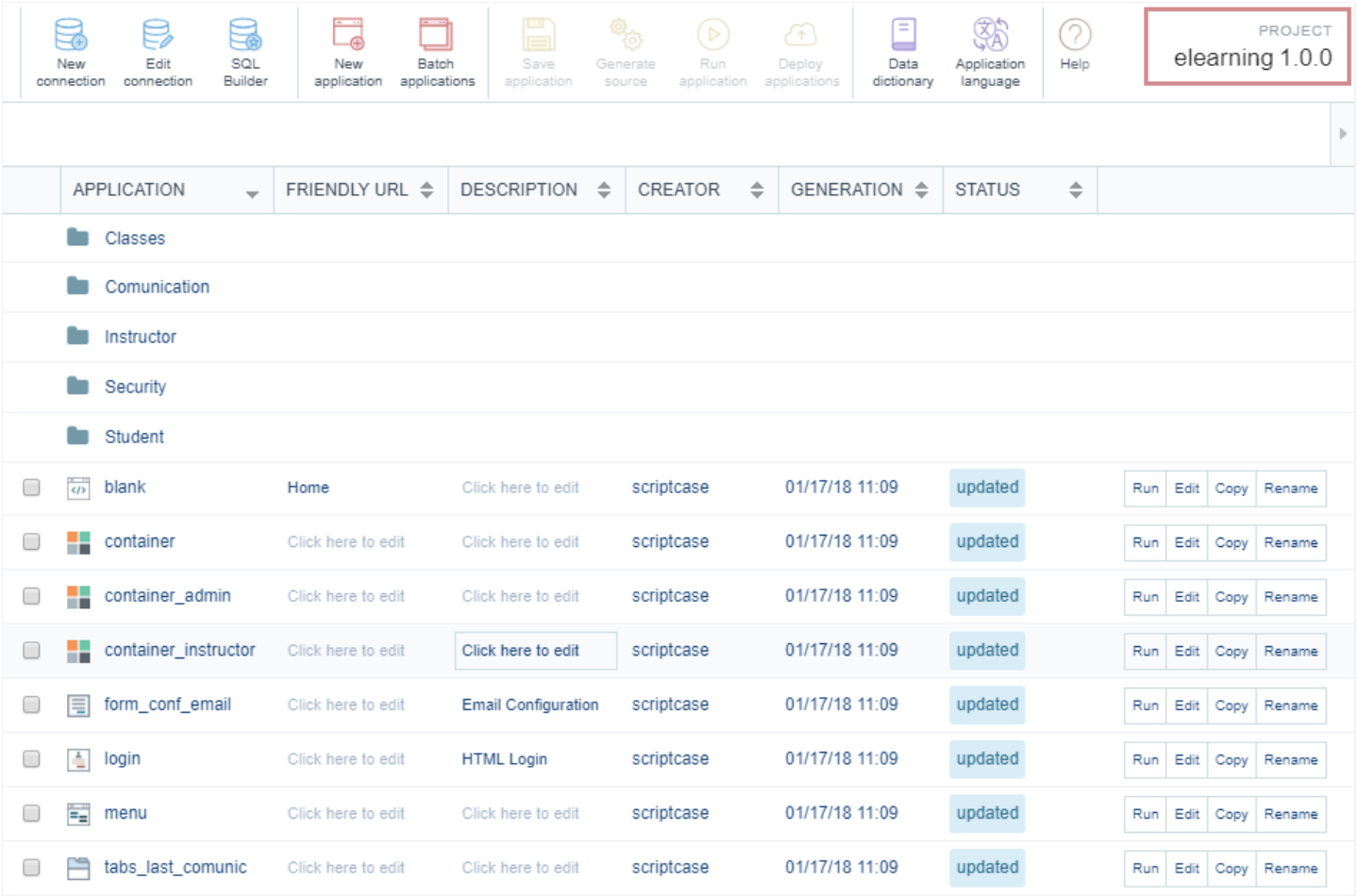
This option redirects to the Project List page, without closing your current project. To return to the application list, just click on the home icon. If you access another project, your current project is closed

Access **Project > Open Project** or click on the Open Project icon on the toolbar.





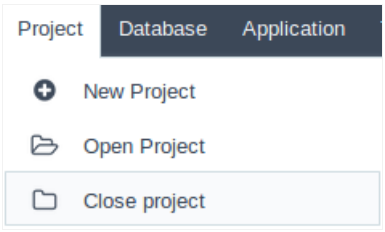
Next, you will visualize the project list. The current project appears on the upper right corner, as shown in image below:



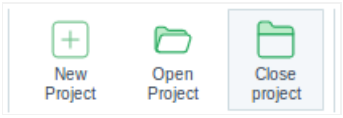
Close Project

Closes the current project redirecting to the the Project List.

Access **Project > Close Project**



Or simple click on the Close Project Icon on the toolbar.

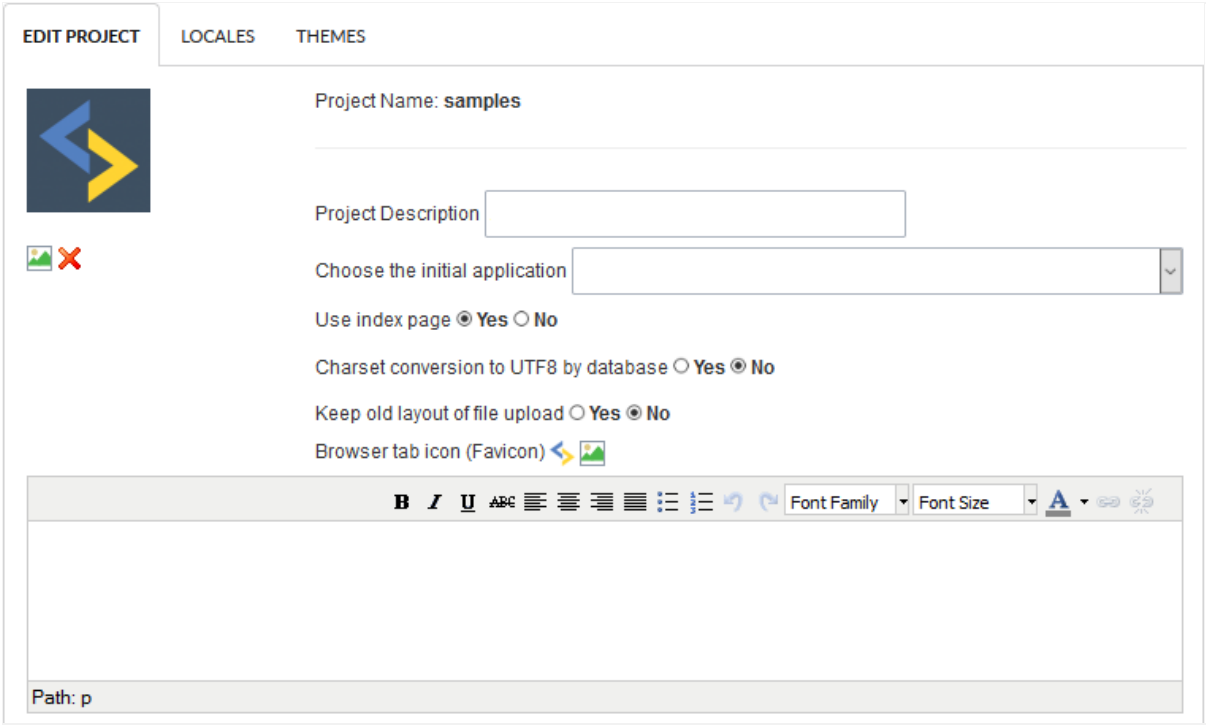


Properties

This option is only available when accessing a project. It allows the editing of the project's information.

Edit Project

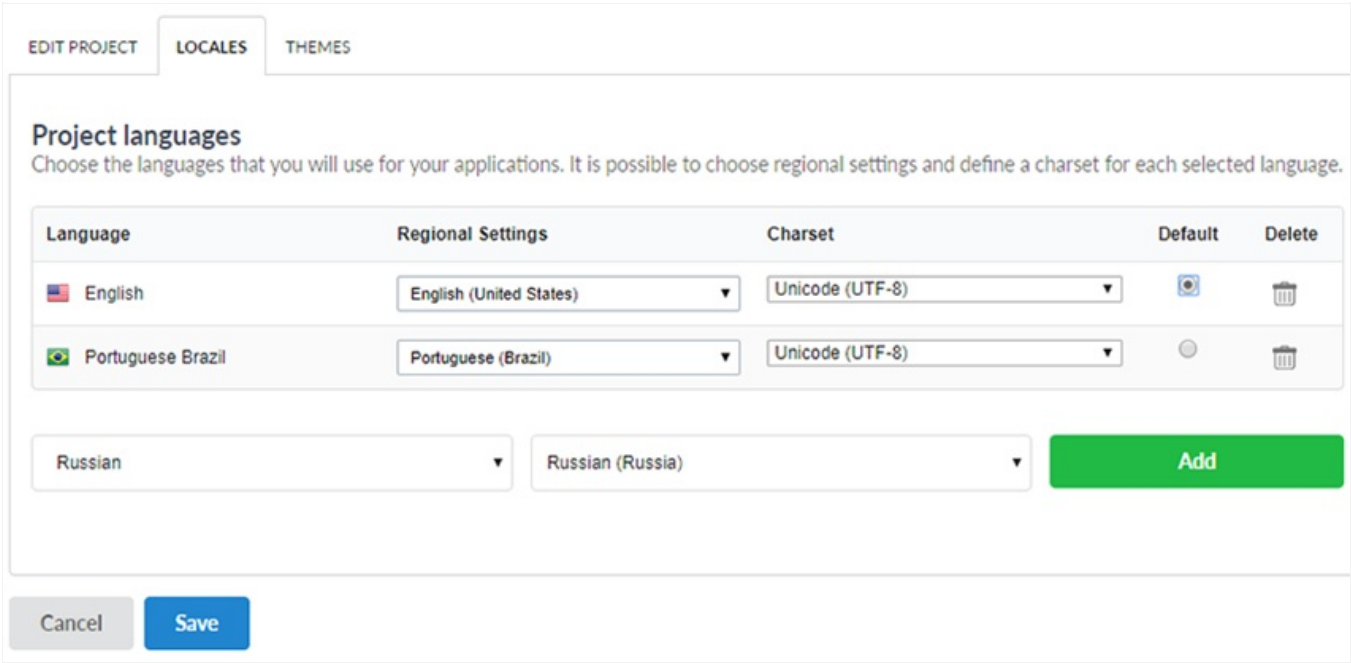
We can edit some of the project information, like the description and the use of the index.



- **Project Name** - The name of the project informed on its creation. It is not possible to change the project name after its creation.
- **Project Description** - Description informed on the project creation. Differently from the project name, it is possible to change it using this interface.
- **Choose the initial application** - Allows you to define the initial project application for the **Run Project** button. So every time you run the project, this application opens automatically, it can be a login screen, for example.
- **Use Index Page** - Allows the use of friendly URLs in the applications. Scriptcase enables this option by default.
- **Charset conversion to UTF8 by database** - With this option enabled, the database will convert the characters sent by the server according to the charset of the database.
- **Keep old layout of file upload** - Option defined for users who do not use the slash symbol (/) when creating a subdirectory, in the image and document type fields. In this scenario, files were stored directly in the "file" directory. If your file structure is like this, it is necessary to enable this flag. In the latest versions, images and documents are being saved in the directories informed in the installation, that is, "file / img" for images and "file / doc" for documents.
- **Project Information** - Detailed description of the project.

Locales

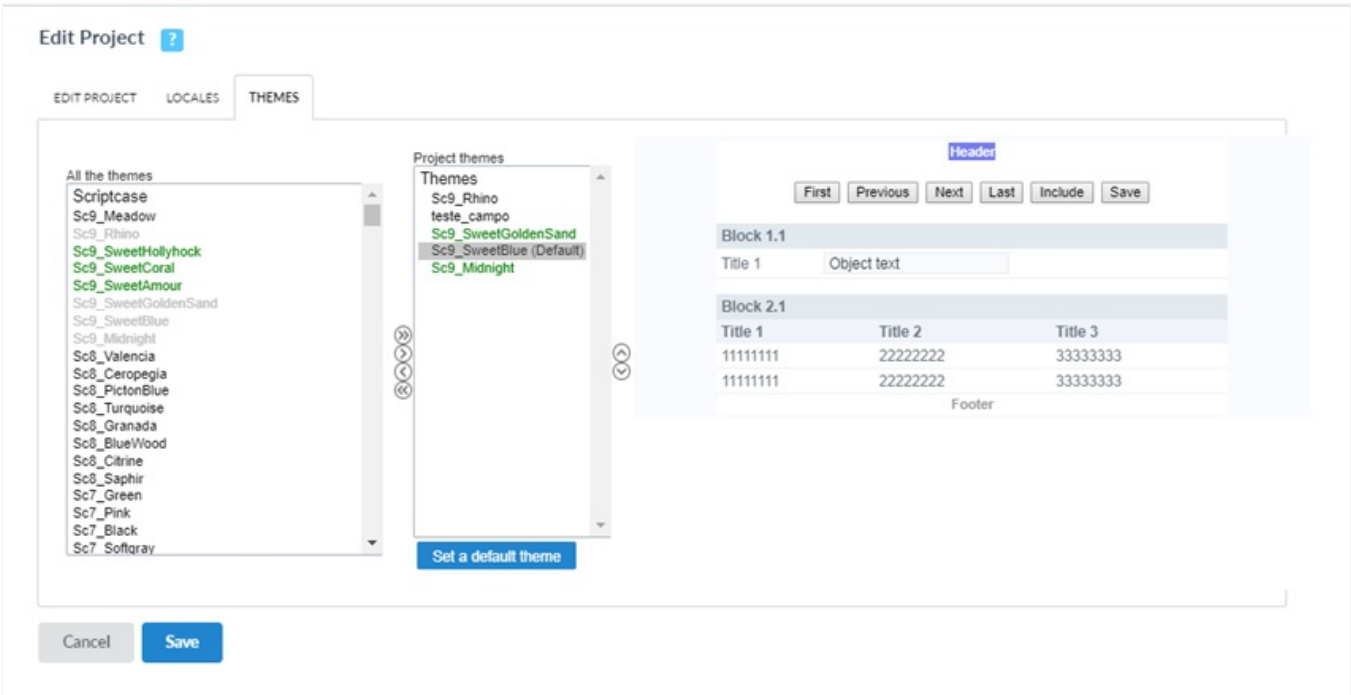
Define the languages for the project applications. The languages are select as you create your project, you can edit this option to add or remove languages anytime.



- **Language** - List of languages from the project.
- **Regional Settings** - Standard regional settings for the selected language. Regional Settings allows you to define some parameters of monetary units, dates, and numbers according to the country or region where your application.
- **Charset** - Encoding used by the selected language.
- **Default** - Default a language for the project. All the applications created are executed on the default selected language.
- **Trash Can** - Removes the language from the project. The removed language won't be available for changing dynamically in runtime.

Themes

It defines the available themes in the project and the default theme used by the applications. To set a default theme for your project, select the theme and click on **Set a default Theme**, the "default" notation appears beside the theme name. In this image, the default theme is **Sc9_Rhino**.



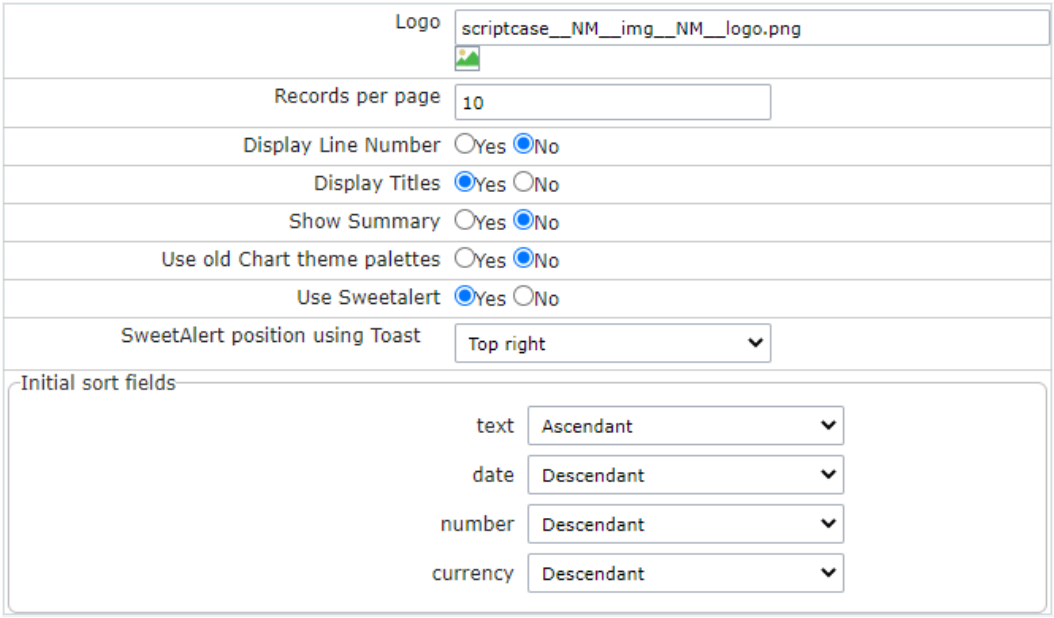
- **All Themes** - List of all the themes from ScriptCase.

- **Project Themes** - List of available themes within the project.

Default Values

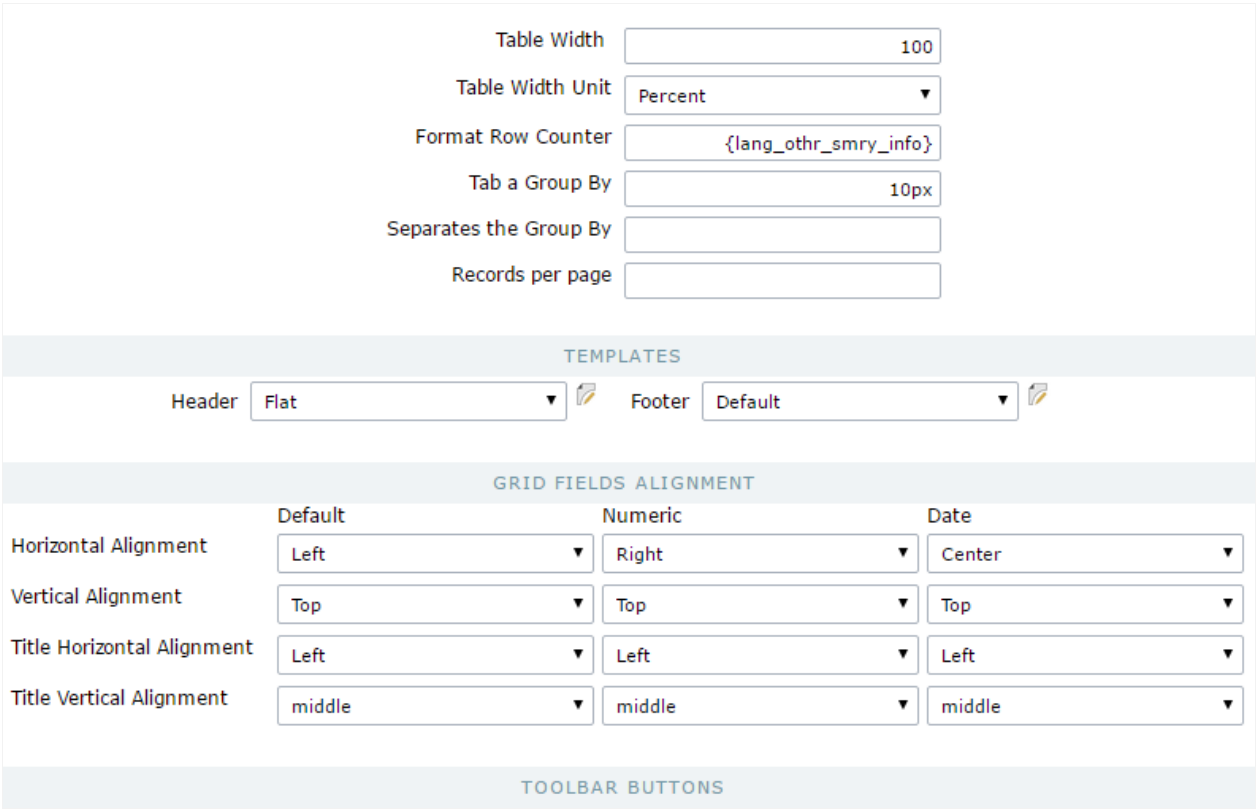
With this configuration, it is possible to standardize the creation of the applications of the current project, increasing productivity in the development of the applications.

Common Settings



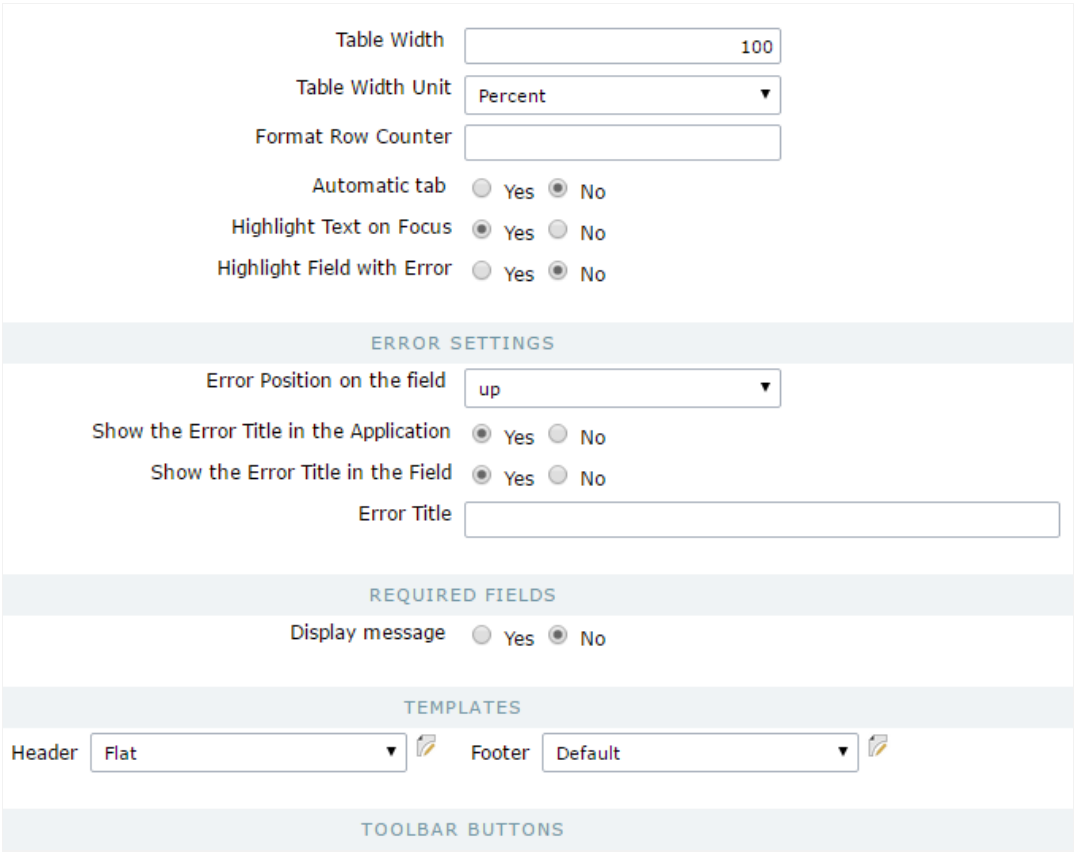
- **Logo** - The Favicon that is going to use with the project. Customize the project using the same Favicon in all the applications.
- **Records per page** - Amount of records per page.
- **Display Line Number** - Defines the display of the line number existing in each application.
- **Display Titles** - Displays the title of the application in execution.
- **Show Summary** - Displays the number of records on the page and the total of the records in the application. This option executes only if the line option in the Grid toolbar is disabled.
- **Use old Chart theme palettes** - Defines the use of the old scheme of colors on the charts
- **Initial sort fields** - Defines the initial sorting of the fields.

Grid



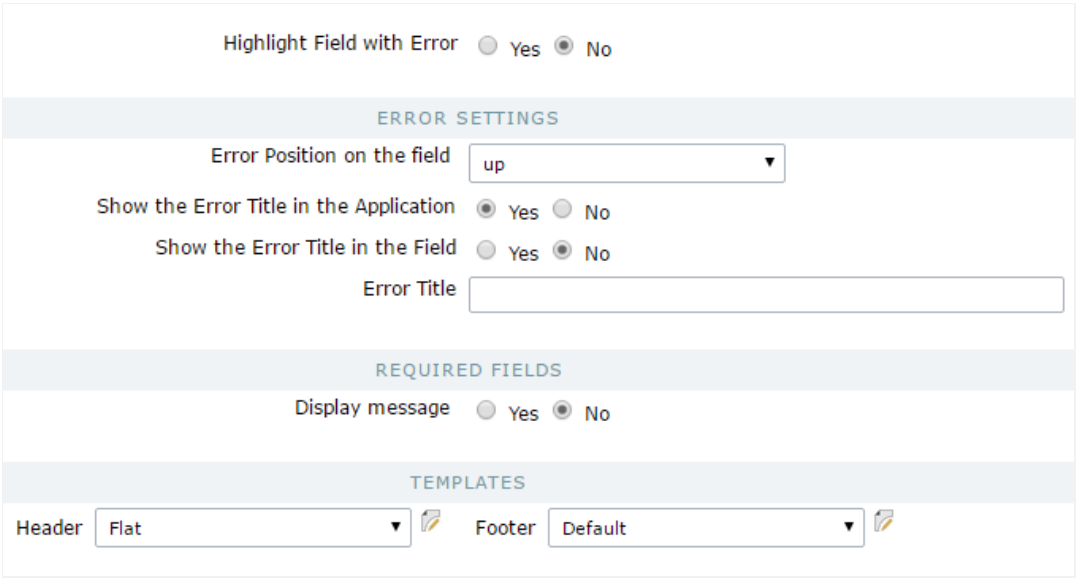
- **Table Width** - Defines the length for the application table. this option is available to select **Pixel** or **Percent** in the table width unit.
- **Table Width Unit** - The unit used to define the width of the application. Automatic (Width automatically defined according to the size of the fields); Pixel (Width defined by pixels, that should be informed in the Form: 800px); Percent (Width defined in percentage, and should be informed in the Form: 80%)
- **Format Row Counter** - Configuration of visualization of the display summary option of the Grid. The display summary option is displayed and executed only if the line option in the toolbar isn't enabled
- **Tab a Group By** - Defines the left margin of the Group By.
- **Separates the Group** - Defines the margin between two Group By.
- **Records per page** - Available values in the Record button, on the toolbar
- **Templates** - Defines the templates used by default on the Header and Footer. It's possible to inform the values to the variables of the selected templates, clicking on the editing icon right beside the field of the template selection.
- **Grid fields alignment** - It's possible to define the default alignment specifically for the fields that contain numbers and dates and the other types of fields.
- **Toolbar Buttons** - Define the buttons that'll be set up by default in the toolbar for the new applications. We can set up the toolbar for the Grid, Detail, and Summary.

Form



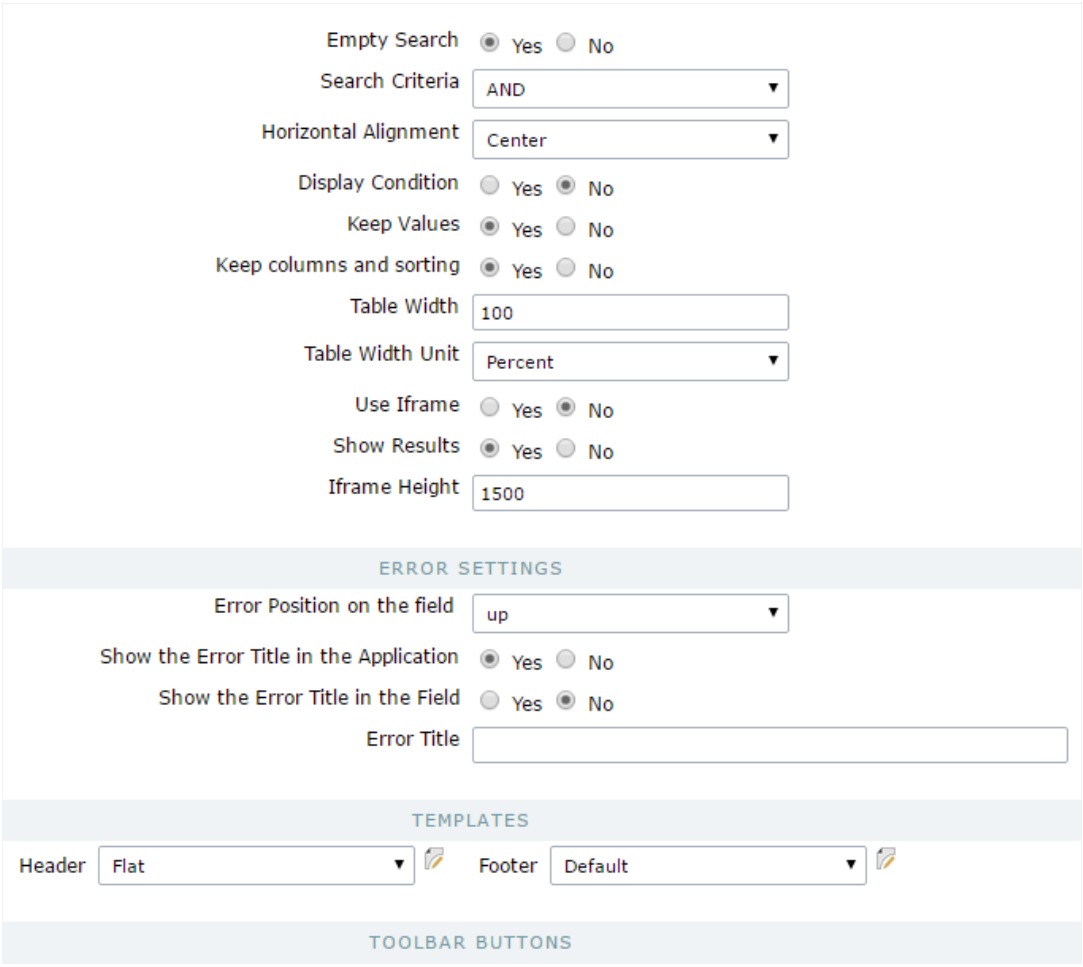
- **Table Width** - Defines the length for the application table. this option is available to select **Pixel** or **Percent** in the table width unit.
- **Table Width Unit** - The unit used to define the width of the application. Automatic (Width automatically defined according to the size of the fields); Pixel (Width defined by pixels, that should be informed in the Form: 800 pixels); Percent (Width defined in percentage, and should be informed in the Form: 80%)
- **Format Row Counter** - Configuration of visualization of the display summary option of the Form on a horizontal orientation. The Display summary option is displayed and executed only if the line option in the toolbar isn't enabled
- **Automatic tab** - Enables the automatic tab when finishing to inform a field.
- **Highlight Text on Focus** - Allows the content of the fields to be Highlighted when the field is focused.
- **Highlight Field with Error** - When occurring an error on a field, it is also required, for example, the focus will be applied to this field.
- **Error Position on the field** - Position of the field error.
- **Show the Error Title in the Application** - Defines the display of the title of the application error.
- **Show the Error Title in the Field** - Defines the display of the error Title for the field.
- **Error Title** - Defines the error Title.
- **Display message** - Defines the display for the message if the field is required.
- **Templates** - Defines the templates used by default on the Header and Footer. It's possible to inform the values to the variables of the selected templates, clicking on the editing icon right beside the field of the template selection.
- **Toolbar Buttons** - Define the buttons that'll be set up by default in the toolbar for the new applications.

Control



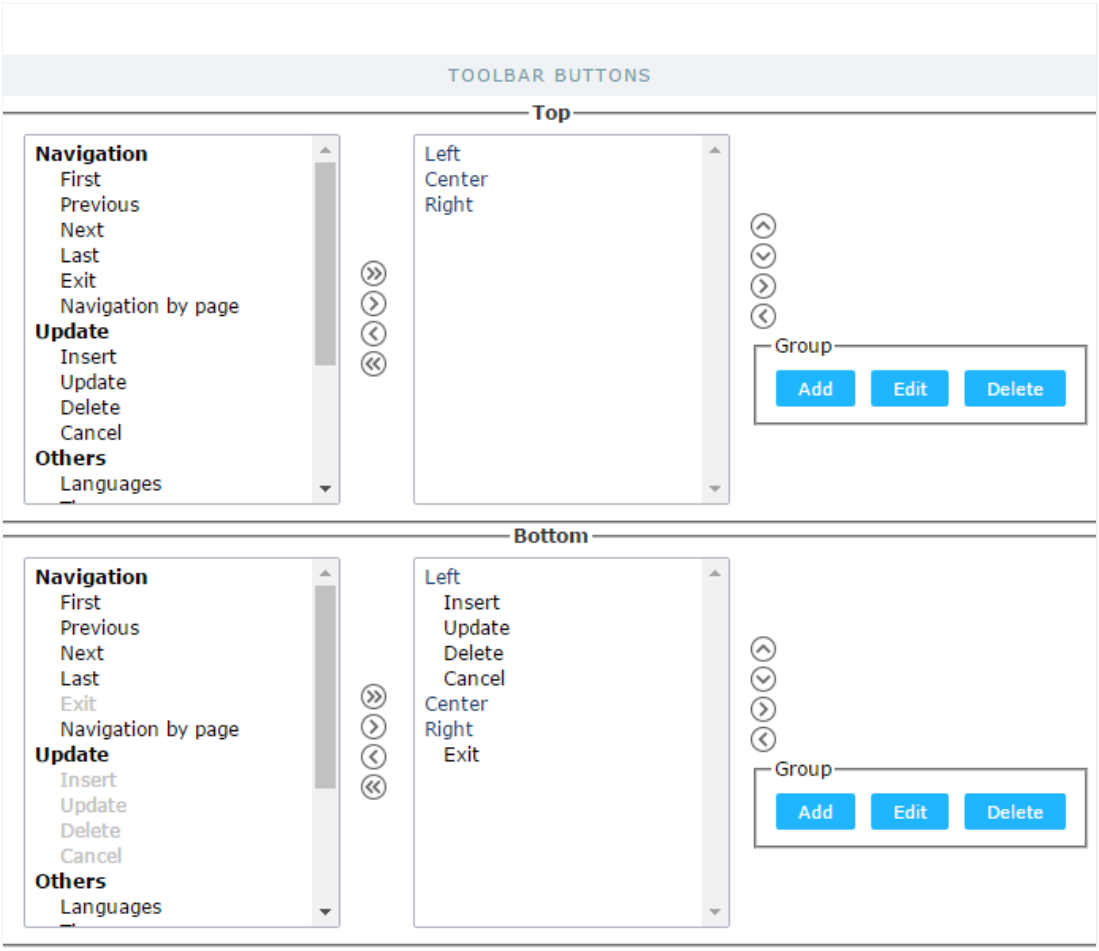
- **Highlight Text on Focus** - Allows the content of the fields to be highlighted when the field is focused.
- **Highlight Field with Error** - When occurring, an error on a field is also required. For example, the focus will be applied to this field.
- **Error Position on the field** - Position of the field error.
- **Show the Error Title in the Application** - Defines the display of the title of the application error.
- **Show the Error Title in the Field** - Defines the display of the error Title for the field.
- **Error Title** - Defines the error Title.
- **Display message** - Defines the display for the message if the field is required.
- **Templates** - Defines the templates used by default on the Header and Footer. It's possible to inform the values to the variables of the selected templates, clicking on the editing icon right beside the field of the template selection.

Search



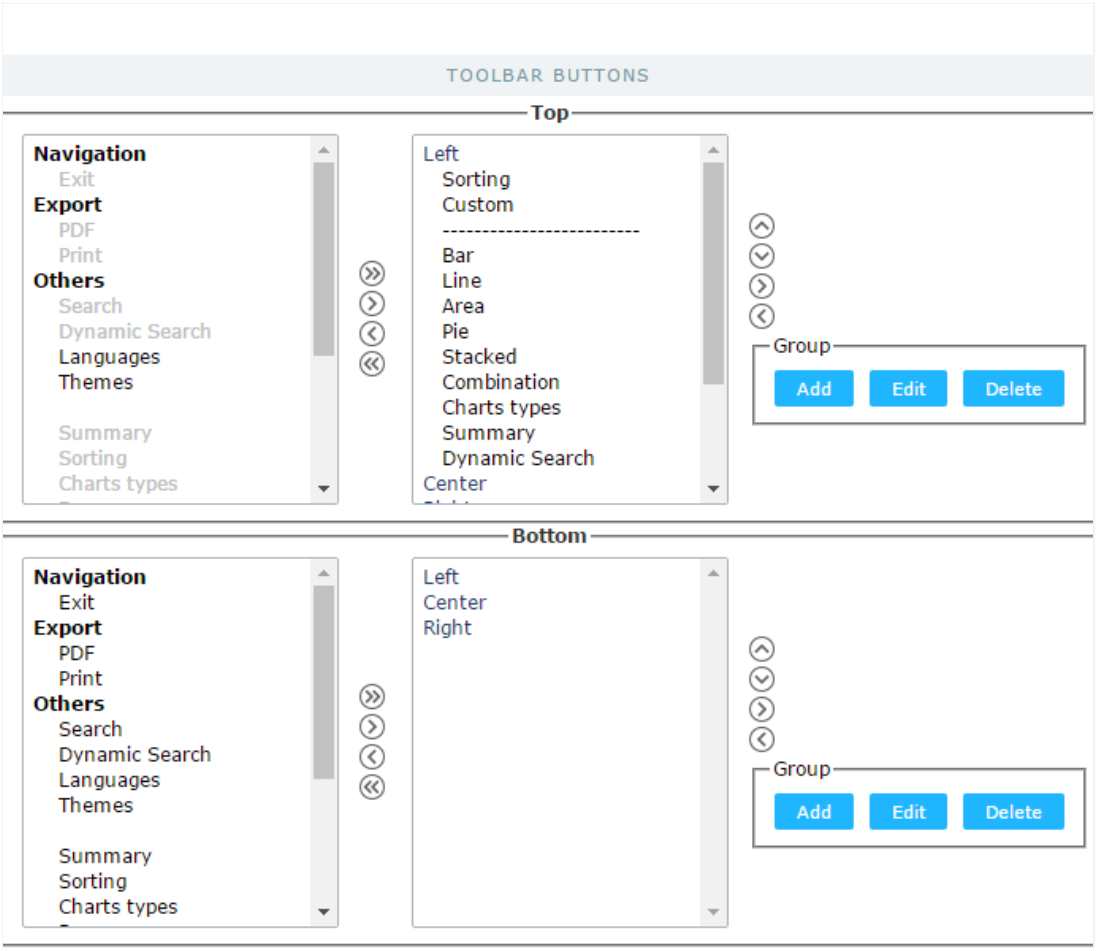
- **Empty Search** - Defines how the search treats the conditions. “AND” (Should satisfy all the conditions so that the search can return some results) “OR” (Should satisfy any condition informed so that the search can return some results)
- **Horizontal Alignment** - Horizontal Alignment of the search.
- **Display Condition** - Displays the condition of the search so that the user can make their choice. (“AND” or “OR”)
- **Keep Values** - Keep the values of the previous searches done when reaccessing the search.
- **Keep Columns and Order Selection** - Preserve the position of the fields and the sorting of the previous search.
- **Table Width** - Defines the length for the application table. this option is available to select **Pixel** or **Percent** in the table width unit.
- **Table Width Unit** - The unit used to define the width of the application. Automatic (Width automatically defined according to the size of the fields); Pixel (Width defined by pixels, that should be informed in the Form: 800 pixes); Percent (Width defined in percentage, and should be informed in the Form: 80%)
- **Use Iframe** - Allows the use and the display of the application searched in the same page within an iframe.
- **Initial State** - Defines how the application is displayed at first when using the search with an iframe, if the application is displayed at first or only the search is displayed.
- **Iframe Height** - Defines the height of the iframe in pixels.
- **Error Position on the field** - Position of the field error.
- **Show the Error Title in the Application** - Defines the display of the title of the application error.
- **Show the Error Title in the Field** - Defines the display of the error Title for the field.
- **Error Title** - Defines the error Title.
- **Display message** - Defines the display for the message if the field is required.
- **Templates** - Defines the templates used be default on the Header and Footer. It’s possible to inform the values to the variables of the selected templates, clicking on the editing icon right beside the field of the template selection.
- **Toolbar Buttons** - Define the buttons that’ll be setup by default in the toolbar for the new applications.

Calendar



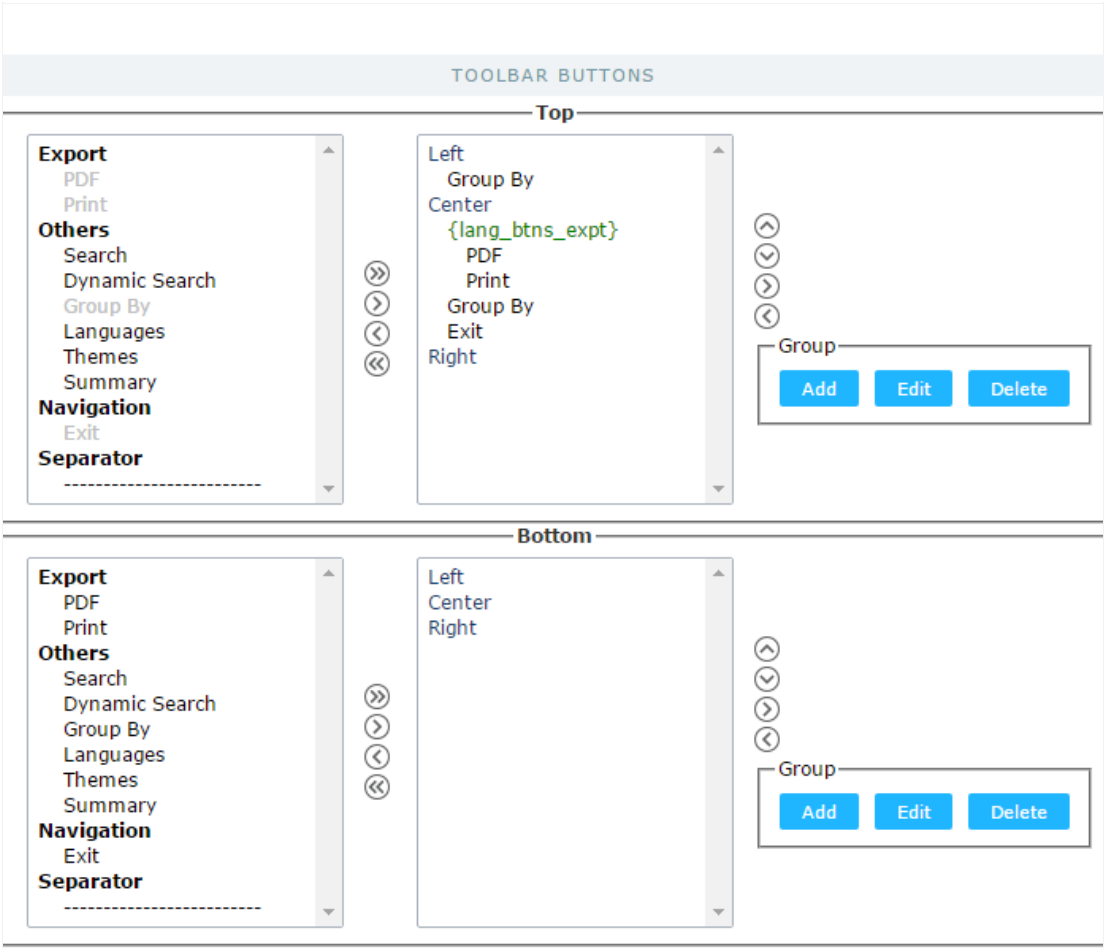
- **Toolbar Buttons** - Define the default buttons in the toolbar for the new applications.

New Chart



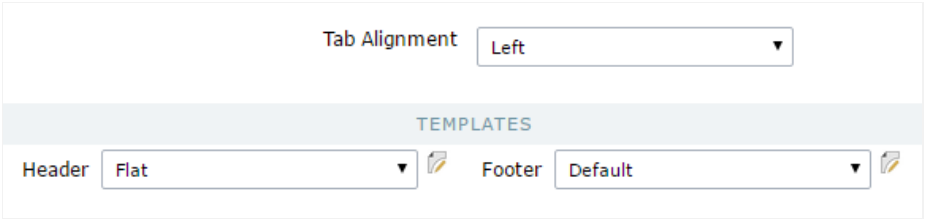
- **Toolbar Buttons** - Define the buttons that will be set up by default in the toolbar for the new applications.

Old Chart



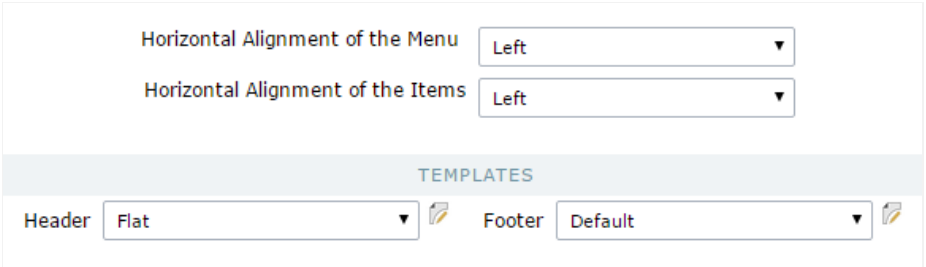
- **Toolbar Buttons** - Define the default buttons in the toolbar for the new applications.

Tab



- **Tab Alignment** - Display alignment of the tabs in the application.
- **Templates** - Defines the templates used by default on the Header and Footer. It's possible to inform the values to the variables of the selected templates, clicking on the editing icon right beside the field of the template selection.

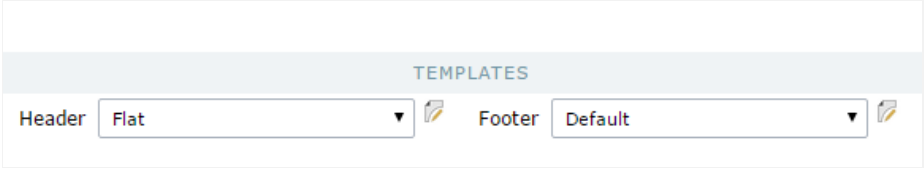
Menu



- **Horizontal Alignment of the Menu** - Menu alignment.

- **Horizontal Alignment of the Items** - Menu item alignment
- **Templates** - Defines the templates used by default on the Header and Footer. It's possible to inform the values to the variables of the selected templates, clicking on the editing icon right beside the field of the template selection.

Dashboard

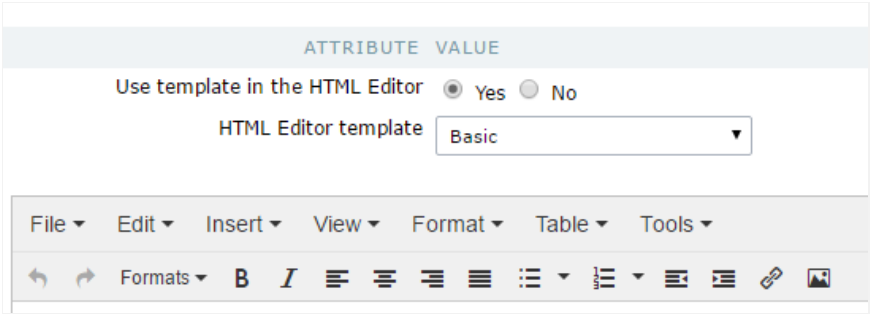


- **Templates** - Defines the templates used by default on the Header and Footer. It's possible to inform the values to the variables of the selected templates, clicking on the editing icon right beside the field of the template selection.

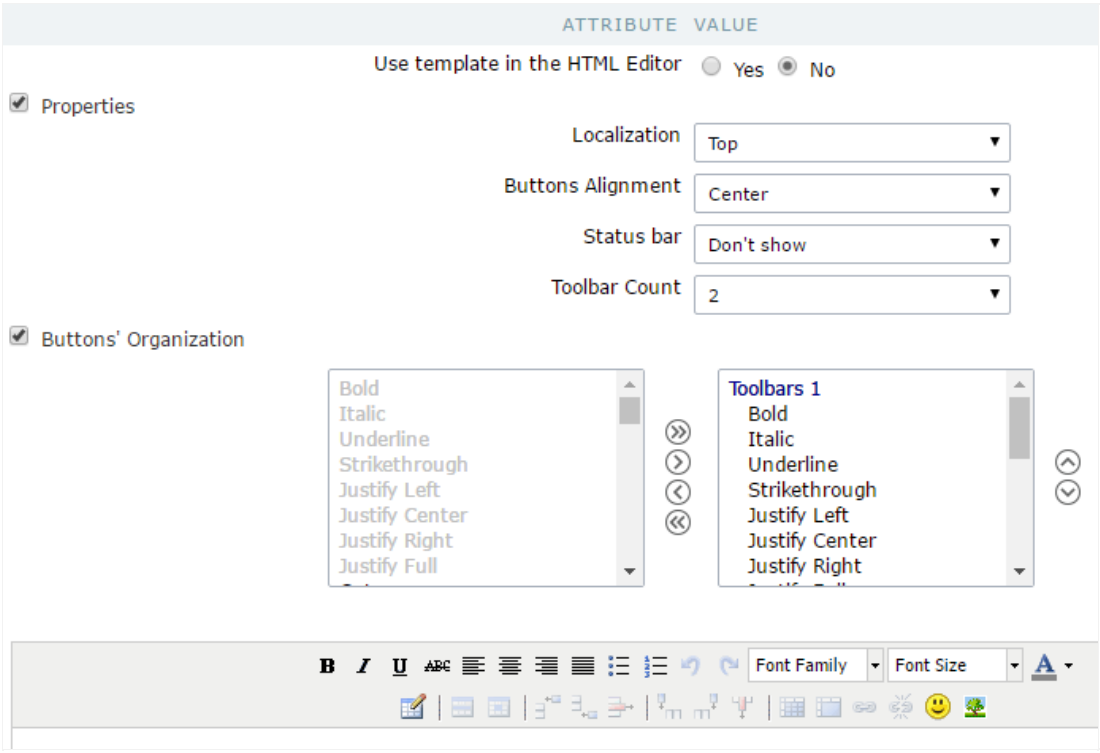
HTML Editor

The HTML Editor has to options of configurations, depending on the option selected on **Use template in the HTML Editor**.

- **Use template in the HTML Editor** - Selecting Yes (Image Below), you use the new templated of the HTML Editor field, these themes should be set up in **Layout > HTML Editor Templates**.
- **HTML Editor template** - Defines the template for editing of the field HTML Editor.
- **Field Preview** - Default Text of the HTML Editor Fields.

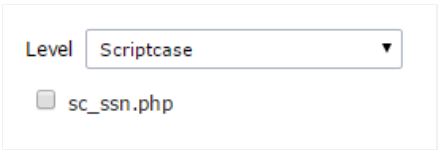


- **Use template in the HTML Editor** - Selecting No (Image Below), you won't be using the new templated of the HTML Editor field. In this case, the options should be set up in the field.
- **Properties** - Defines the display properties.
- **Location** - Defines the position of the items in the field. Top or Bottom
- **Buttons Alignment** - Defines the alignment of the items in the field. Left, Center, Right.
- **Status bar** - Position of the Status Bar.
- **Toolbar Count** - Amount of the bars to organize the items.
- **Buttons' Organization** - Set the available items in the bars.



Internal Libraries

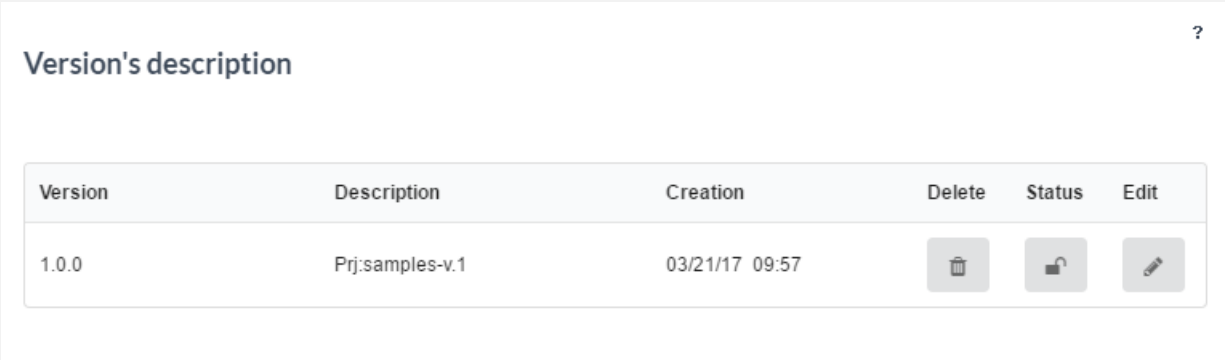
Define the libraries available in the project.



- **Level** - Access level of the library.
- **List of Libraries** - List of all the libraries available for use in the project.

Version History

Lists all the versions of the current project. The Version History page manages the versions of the project, created in the option of version incrementation, here we can return to earlier versions, open a previous version or delete created versions.

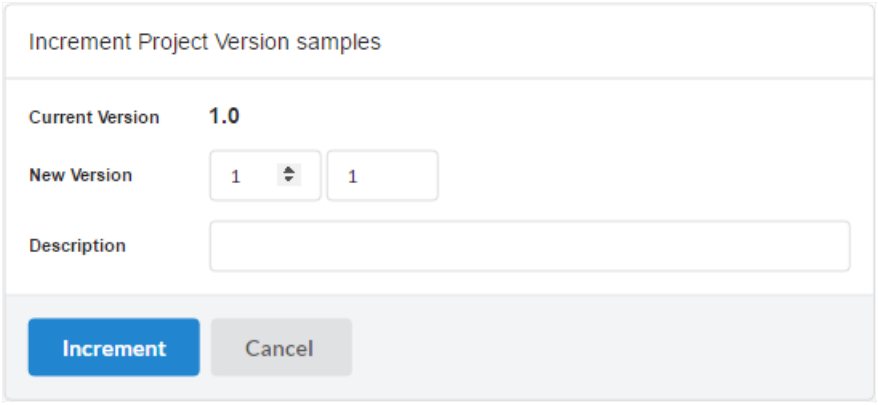


- **Version** - Project Version.
- **Description** - Project version description.
- **Creation** - Date of the version creation.
- **Delete** - Delete the selected version of the project and all its applications. Applications from the other versions are not affected.
- **Status** - Defines if the version of the project is opened or closed.
- **Edit** - Allows to edit the selected version of the project. This option is available for the version that have the status Open.

Version Incrementing

Allows the control over the project in development, separating it in versions. The project version control allows you to protect the functionalities existing in the system, before doing any important modifications, by creating a new version of the current project.

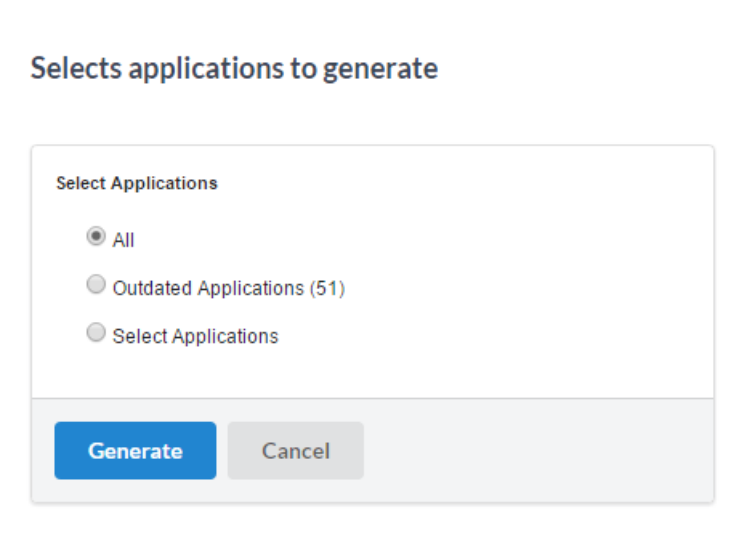
The version increment allows that only the latest version of the project to be increased.



- **Current Version** - Informs the current version of the project.
- **New Version** - Defines the number of the new version of the project.
- **Description** - New project description.

Generating Source Code

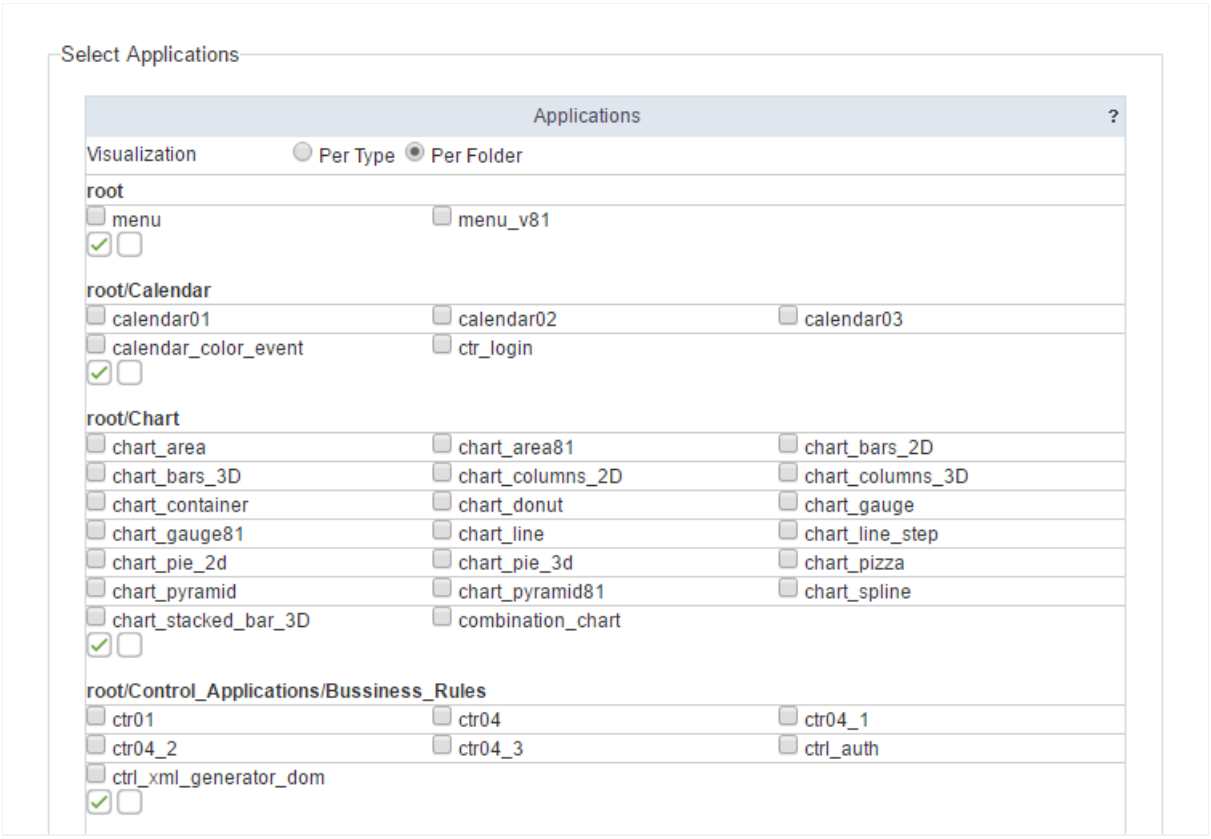
Allows you to generate the source code for all the applications in the project, select the chosen application, or generate the source code for the outdated ones.



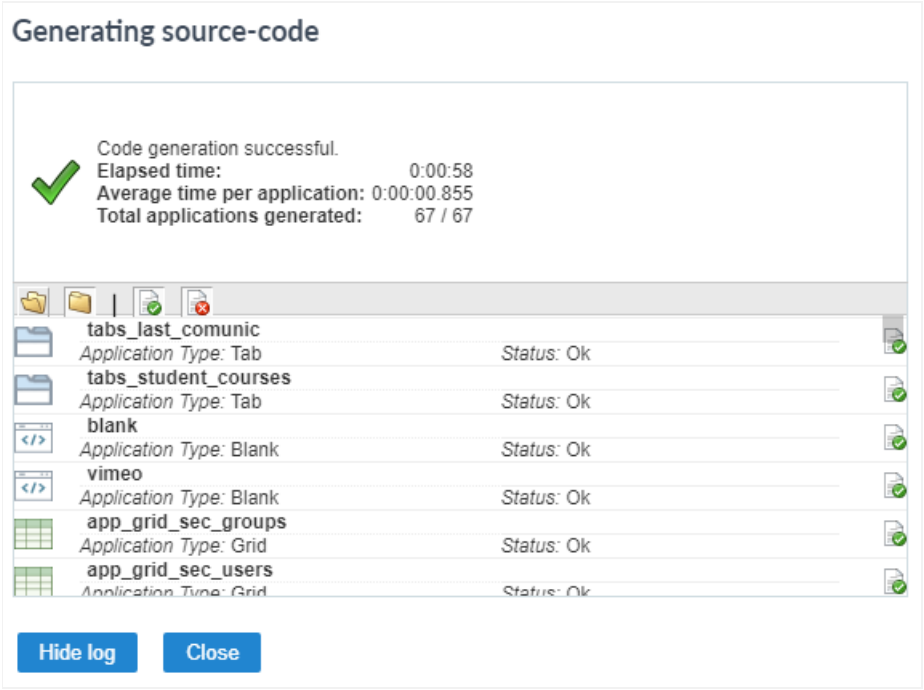
- **All** - Generates the source code for all the applications in the current project when selecting this option, you are redirected to the page where the code is generated, where you can also see the log.
- **Outdated Applications** - Generates the source code only for the outdated applications (recently modified applications that weren't generated yet), when selecting this option, you will be redirected to the page where the code will be generated, where you can see the log.
- **Select Applications** - This option allows you to select the applications that you want to generate the source code, when selecting this option, you will be redirected to the application selection page.

Selecting Applications

All the applications from the project are listed here, and you should select the applications that you want the source code to be generated. The application can be listed in two ways, using the filter to list by Application types or by folders, as shown below.



After selecting the application that will be generated, you will be redirected to the page where the code will be generated, where you can also see the log.



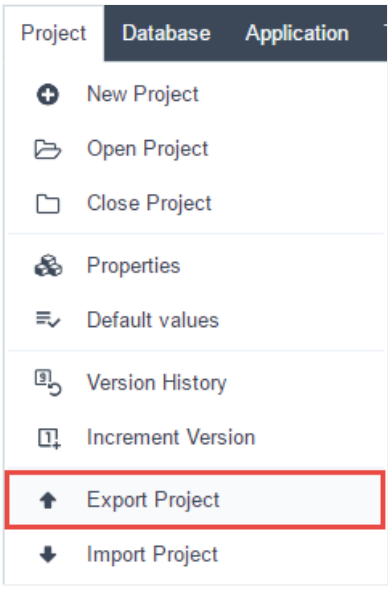
We can see in the highlighted folder of the image some filter options.

- **Open Folder Icon** - Allows possible errors to be displayed when generating the code.
- **Closed Folder Icon** - Allows to hide the possible errors when generating the code.
- **Document ok** - Displays only the applications that haven't had any problems generating the code.
- **Document error** - Displays only the applications that has had any problems generating the code.

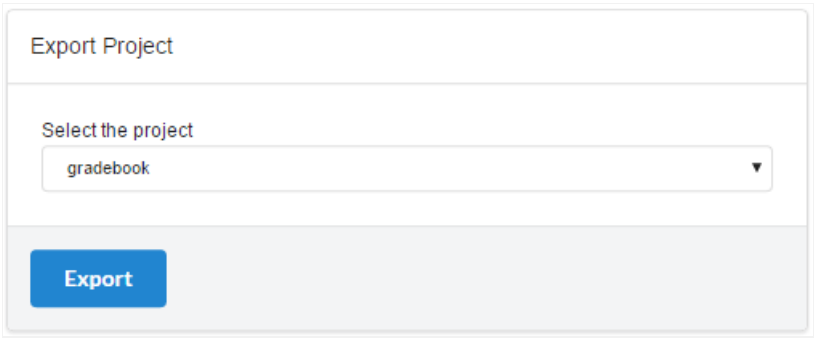
Export Project

This option allows you to create a .zip file with all the applications from the current project and all its required files.

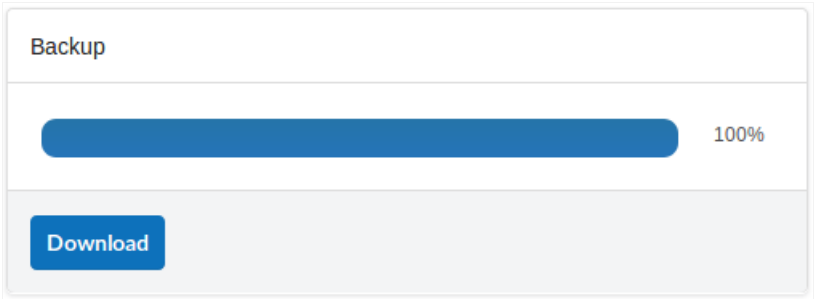
Access **File > Export Project**.



Next, select the project that you want to export.



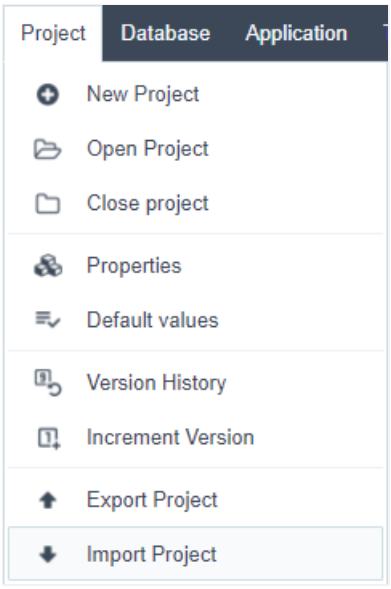
When finishing the process, the file will be available for download.



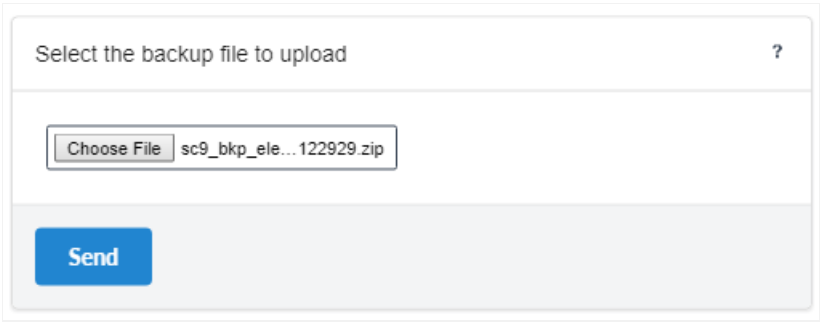
Import Project

Allows you to import a project exported by ScriptCase, with all the required files.

Access **File > Import Project**.

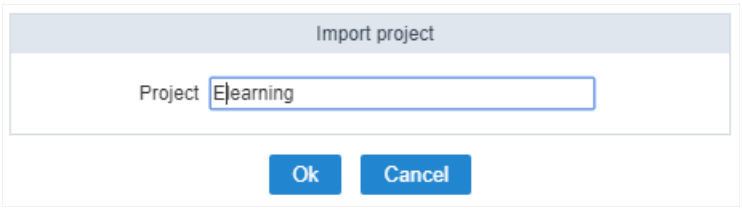


Next, we need to select the project file that we want to import.

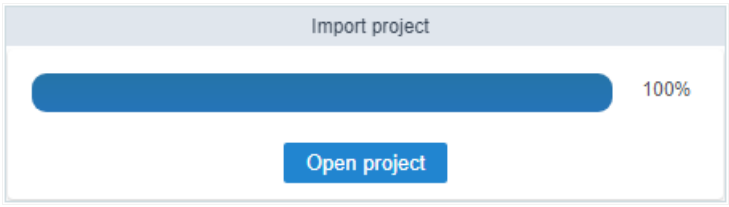


After finishing the process, we should inform the name of the project that is being imported.

In case there is a project with the same name in your ScriptCase, the project that is being imported will overwrite the existing project.



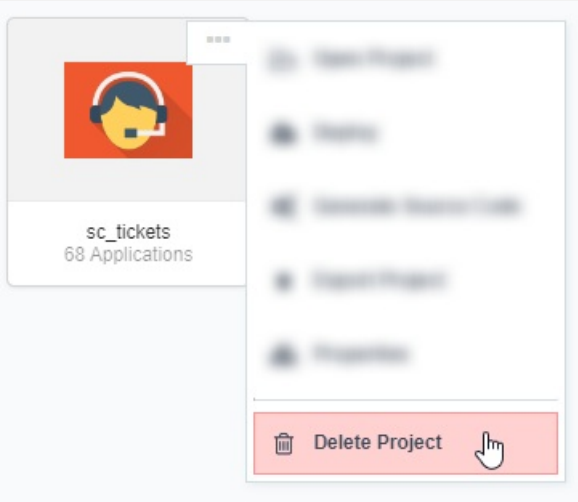
After finishing the import, we can open the project.


Import project
100%
Open project

Deleting a Project

The project can be deleted by **Floating menu** available in the projects interface. Click here for more menu details.

Floating menu at project list



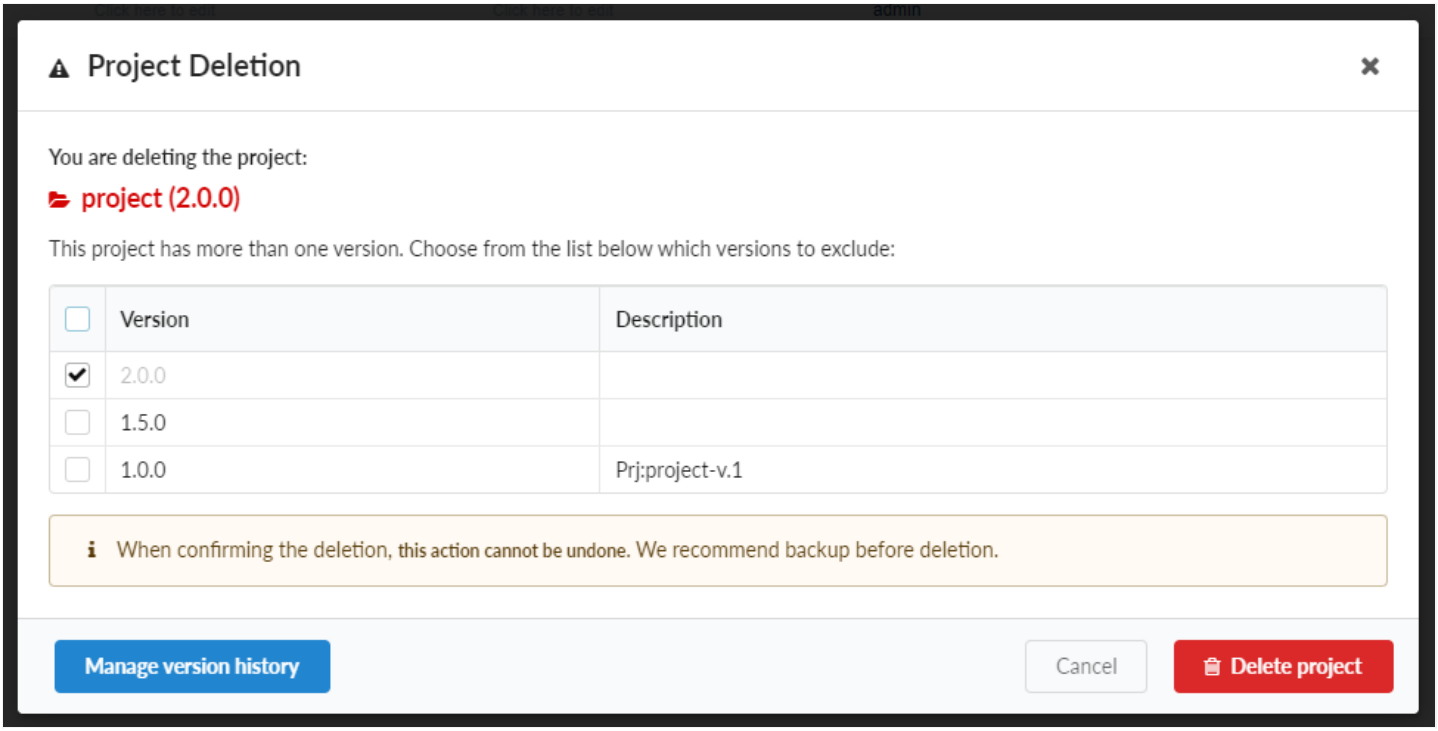
Or can be deleted by menu **Project > Delete project**, in this case will be necessary to open the project.

Project Menu

Delete confirmation

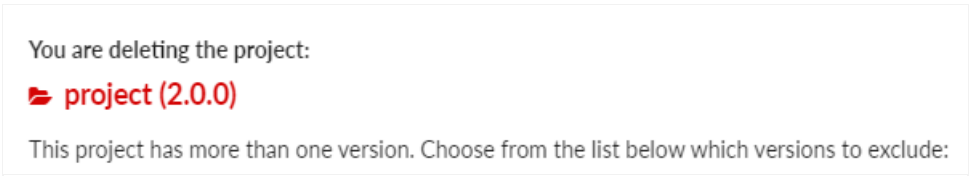
Now will open a pop-up to confirmation the action.

Delete a project can't be undone, so we recommend doing a backup to avoid a definition lost of the project.



On this screen we will have some important information to delete a project.

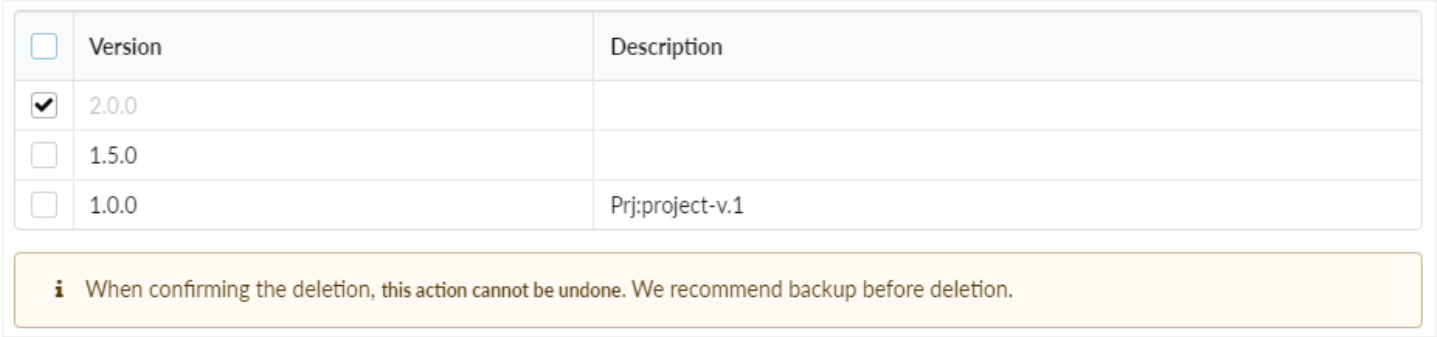
Confirmation of project name and version



In case the project has more then one version, configured at increment Project, it will show them with the possibility to delete or not.

After deleting a project, we have to delete the open version, unless it has the close status. Click here for more details about it.

List of project versions

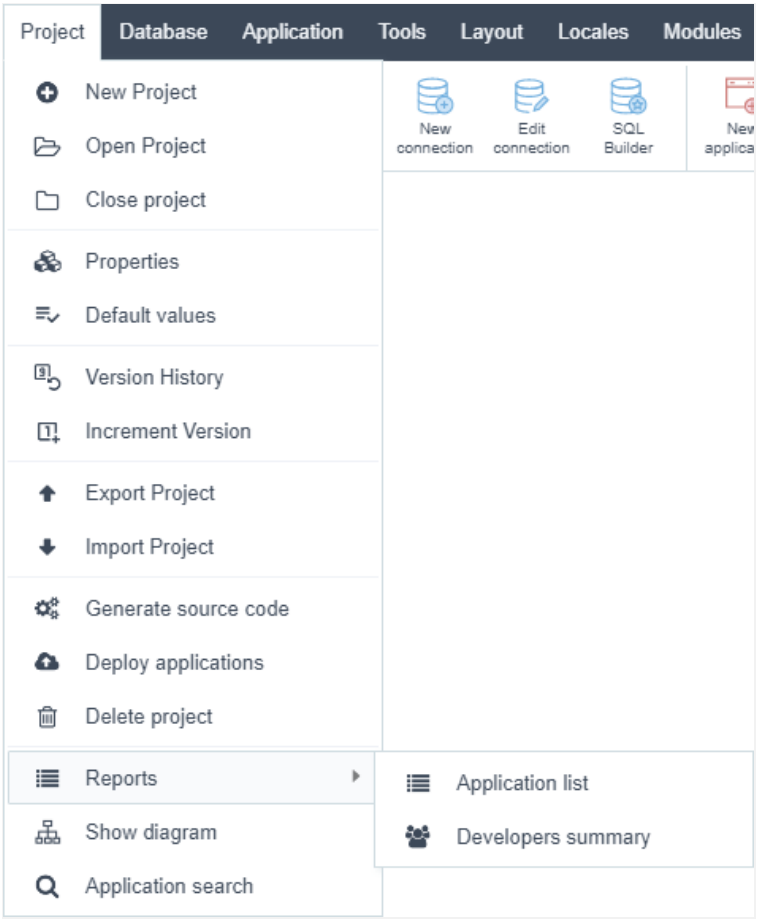


Related Videos ▷

Reports

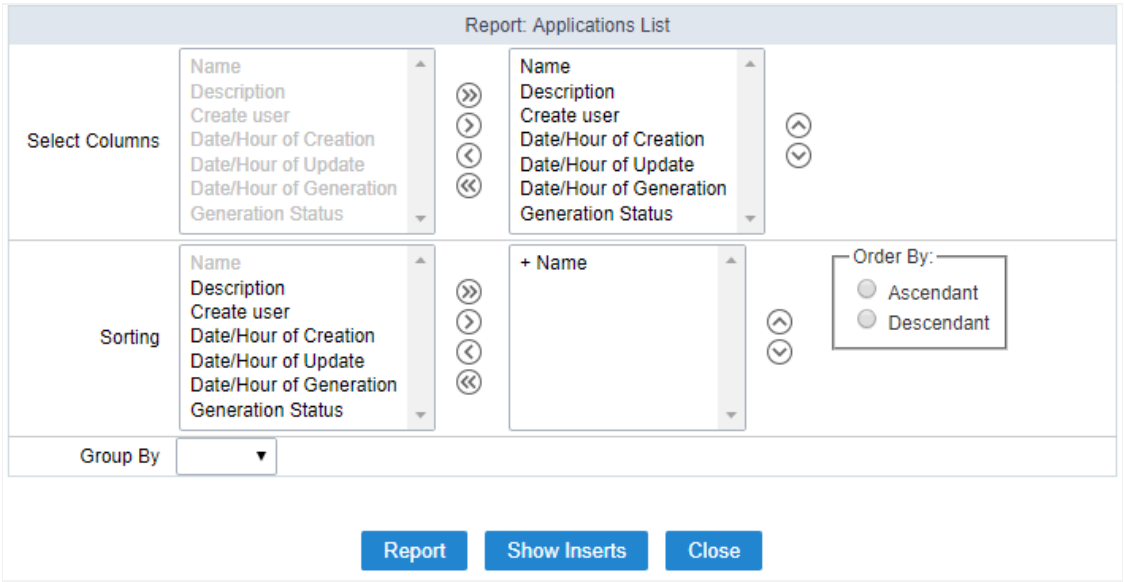
We offer two types of reports for projects: Application List and Developers Summary.

To generate the reports, access **Project > Reports**



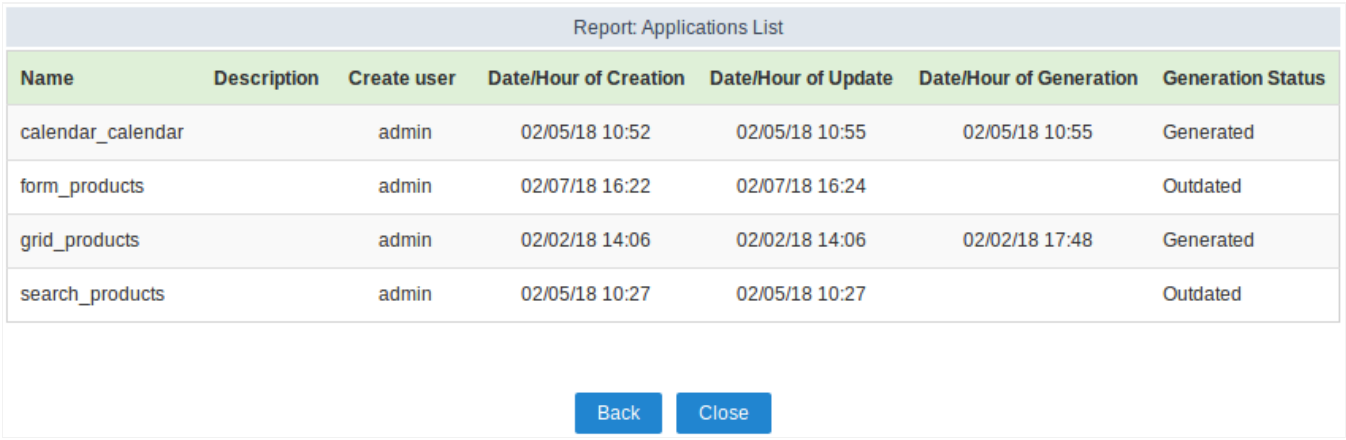
Application List

Displays a list with all the existing applications in the current project and the **SQL INSERT Queries** of the tables. To generate the report, first you need to define which information you want to be available.



- **Select Columns** - Defines the columns that will be part of the report.
- **Sorting** - Defines the fields and the sorting that will be available.
- **Order By** - Defines the initial sorting for the report. It will use **Ascendant** by default if not informed.
- **Group By** - Defines the grouping of the report, by type (application) or by folder (Project's File Directory).

After finishing the configurations, click on **Report**.



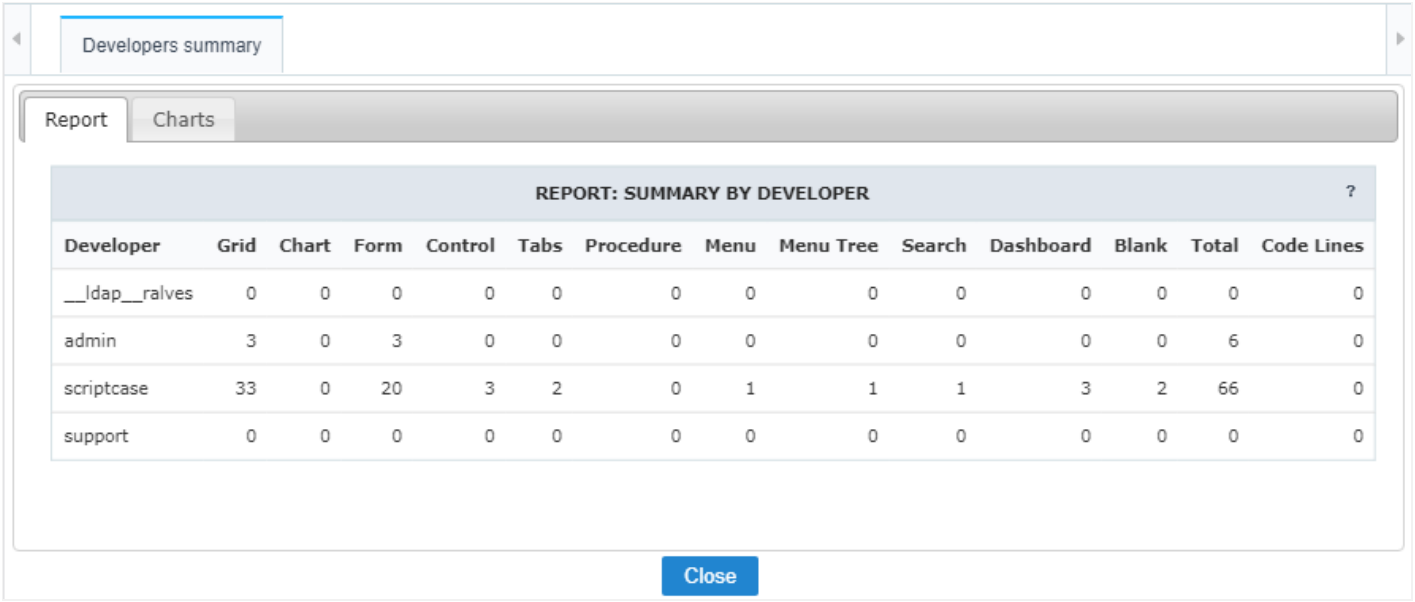
Report: Applications List

Name	Description	Create user	Date/Hour of Creation	Date/Hour of Update	Date/Hour of Generation	Generation Status
calendar_calendar		admin	02/05/18 10:52	02/05/18 10:55	02/05/18 10:55	Generated
form_products		admin	02/07/18 16:22	02/07/18 16:24		Outdated
grid_products		admin	02/02/18 14:06	02/02/18 14:06	02/02/18 17:48	Generated
search_products		admin	02/05/18 10:27	02/05/18 10:27		Outdated

Back Close

Developers Summary

Generates a Summary with every action each developer had done, showing the total applications and their types and the total of code lines that the user wrote in the applications.



Developers summary

Report Charts

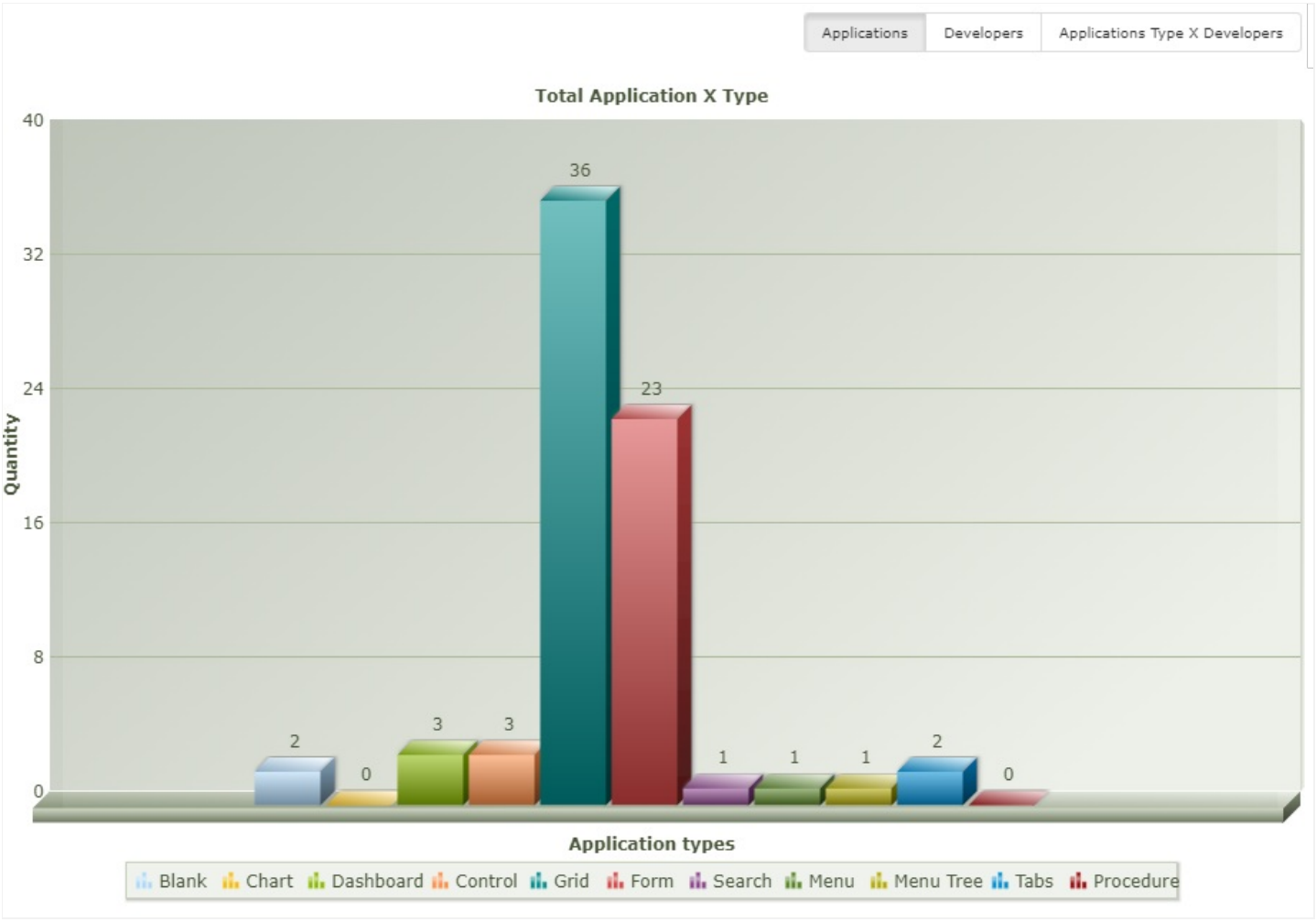
REPORT: SUMMARY BY DEVELOPER ?

Developer	Grid	Chart	Form	Control	Tabs	Procedure	Menu	Menu Tree	Search	Dashboard	Blank	Total	Code Lines
__ldap__ralves	0	0	0	0	0	0	0	0	0	0	0	0	0
admin	3	0	3	0	0	0	0	0	0	0	0	6	0
scriptcase	33	0	20	3	2	0	1	1	1	3	2	66	0
support	0	0	0	0	0	0	0	0	0	0	0	0	0

Close

This same report is available in Charts, with three types of visualization.

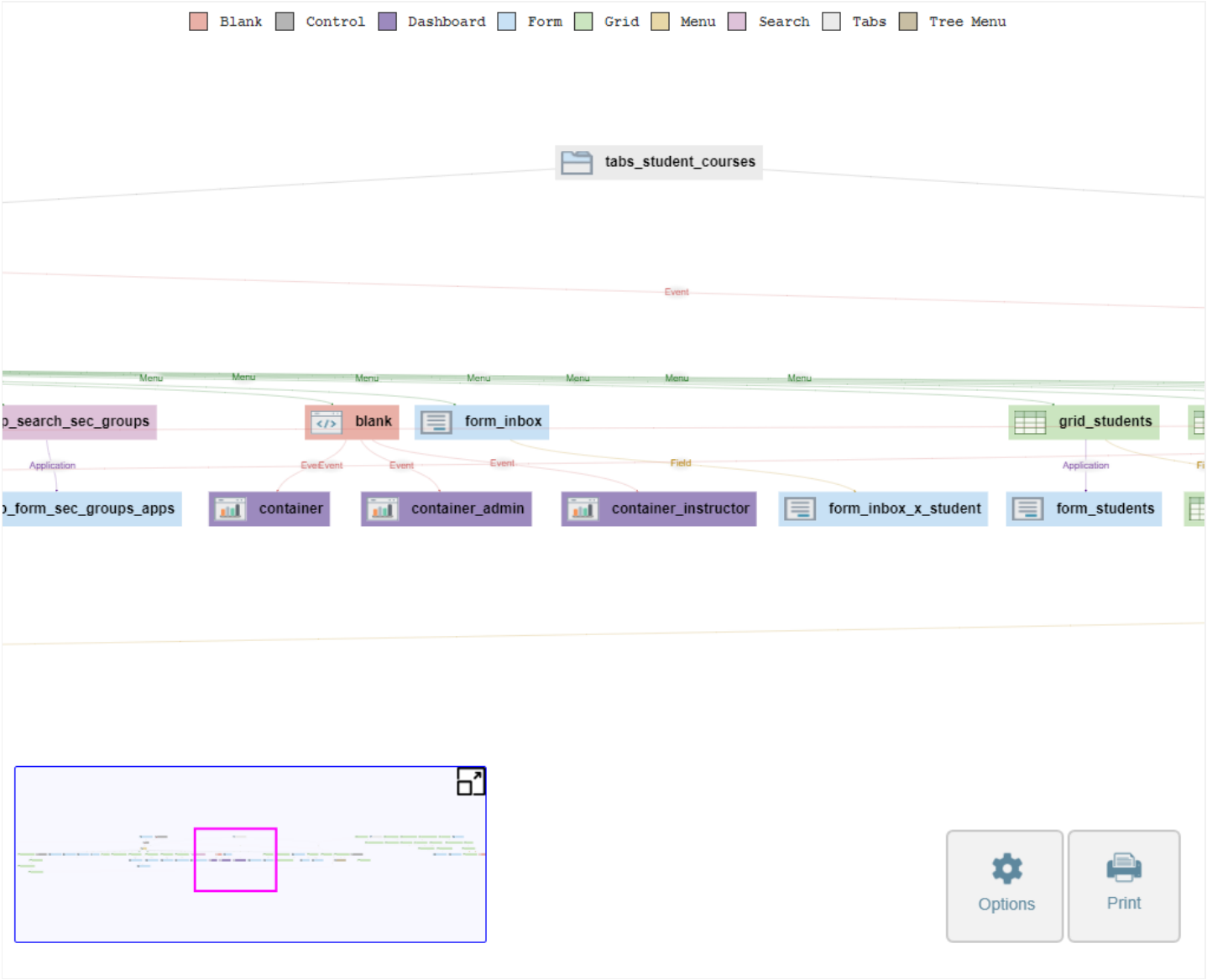
- **Applications** - Total of applications in the project grouped by their type.
- **Developers** - Total of application created by developer in the project.
- **Applications x Developers** - A junction of the other two charts.


Applications
Developers
Applications Type X Developers
Total Application X Type
Quantity
40
32
24
16
8
0
2
0
3
3
36
23
1
1
1
2
0
Application types
Blank
Chart
Dashboard
Control
Grid
Form
Search
Menu
Menu Tree
Tabs
Procedure

Showing Diagram

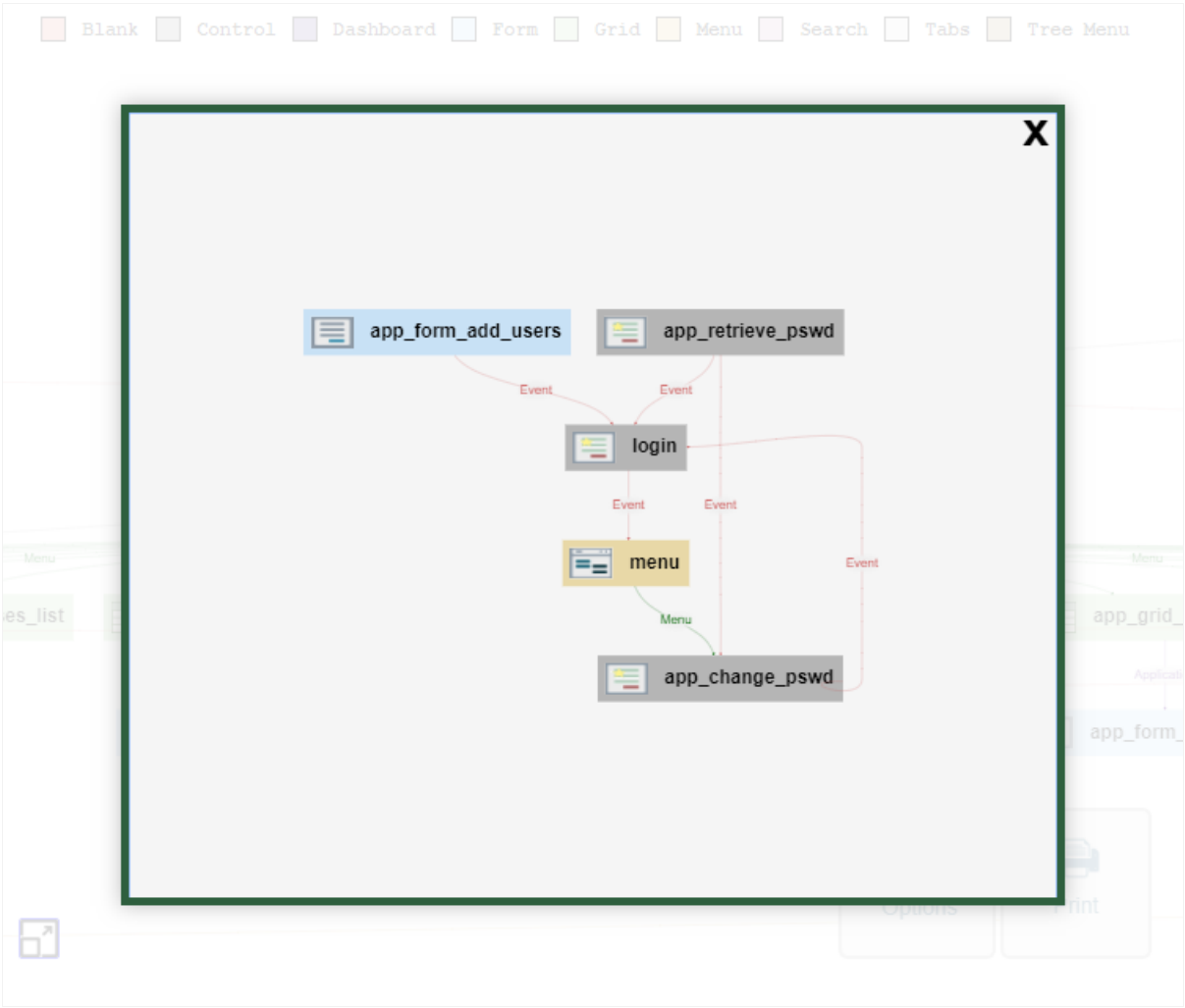
Generates a diagram of relationships between applications in the project. On this diagram, you can see in a clear way which applications are related and which is the method used to create the relationship.

On this example we're showing the relationship of all the applications in the sc_album project.

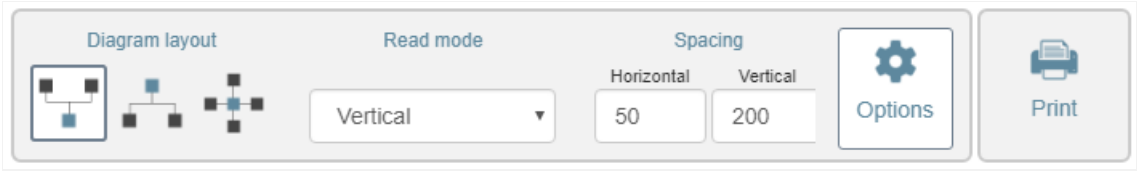


There are some options that can help visualizing the diagram.

We can display a mini map of all the diagrams, to help with the display. In order to visualize it, click on the icon on the lower left side of the diagram.



There are other options of views, instead of the mini map, as shown below.



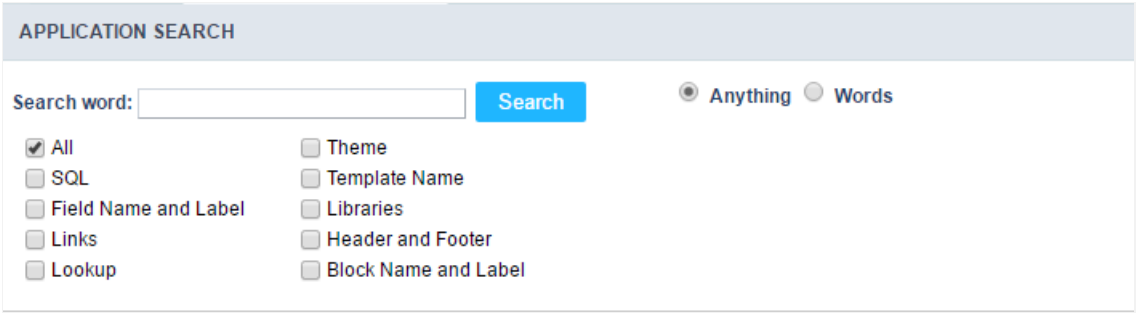
- **Diagram Format** - Build the Diagram according to the ties positioning.
- **Reading Mode** - Defines the perspective of reading for the diagram when it's created.
- **Space** - Defines the space between the columns(Horizontally) and the layers(Vertically).

Application Search

It performs a search in the project with all the content informed by the search itself.

The location on which the search will look for defined by a checkbox.

At least one of the options need to be selected.



- **All** - Performs a search in all the options listed below.
- **SQL** - Searches only in the SQL areas of the applications.
- **Field Name and label** - Performs a search for the name and label of the field.
- **Links** - Performs a search only for the name of the applications.
- **Lookup** - Searches inky in the lookup area of the fields.
- **Themes** - Searches only for theme names.
- **Template Name** - Searches only for HTML templates names.
- **Libraries** - Searches for internal libraries that contain this specific name.
- **Header and Footer** - Searches only for the Application Titles.
- **Block Name and Label** - Searches only for the name or label of the block informed.

The results will be displayed as shown below.

Application	Item	Value
chart01 (Grid)		
	SQL Select Statement	SELECT orderid, customerid, employeeid, orderdate, requireddate, shippeddate, shipvia, freight, priceorder, shipcountry, shipregion, shipstate, shipcity, shipname, shipaddress, shippostalcode FROM orders
	Grid Title	{lang_header_grid_70_title}
	Detail Title	{lang_header_grid_70_title}
	Blocks	0 Name => grid_orders Text => grid_orders
chart02 (Grid)		
	SQL Select Statement	SELECT orderid, customerid, employeeid, orderdate, requireddate, shippeddate, shipvia, freight, priceorder, shipcountry, shipregion, shipstate,

- **Application** - Name of the applications found that satisfies the search.
- **Item** - Location where the search criteria was found.
- **Values** - Values found that satisfies the search.

Importing a template Project

In this tutorial, you learn how to import a project template from Scriptcase. The project templates contain various samples applications for you to copy or learn how to apply Scriptcase features. You can run all the template projects to test by accessing the website here.

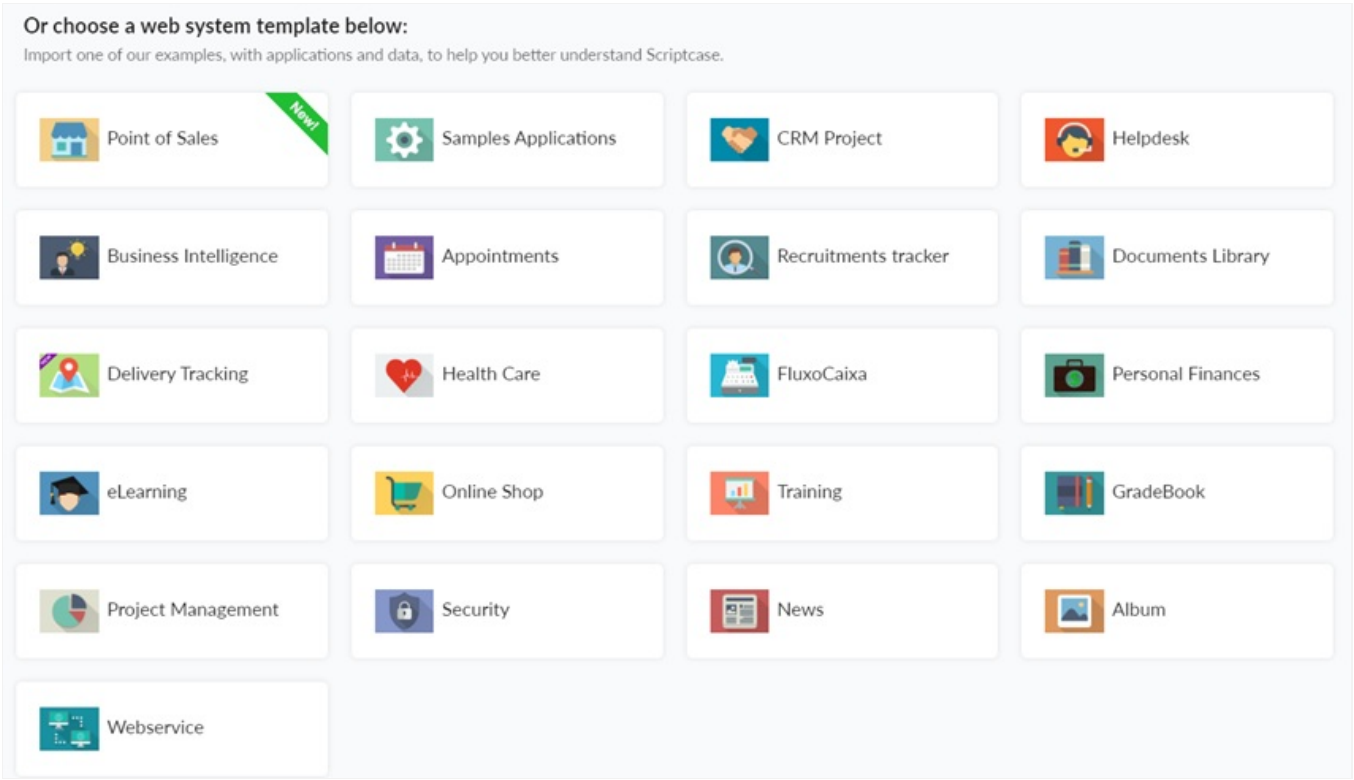
The process of importing a template is divided into two steps, which are detailed below.
The projects created with access permission for the user, appear in

Project List

.

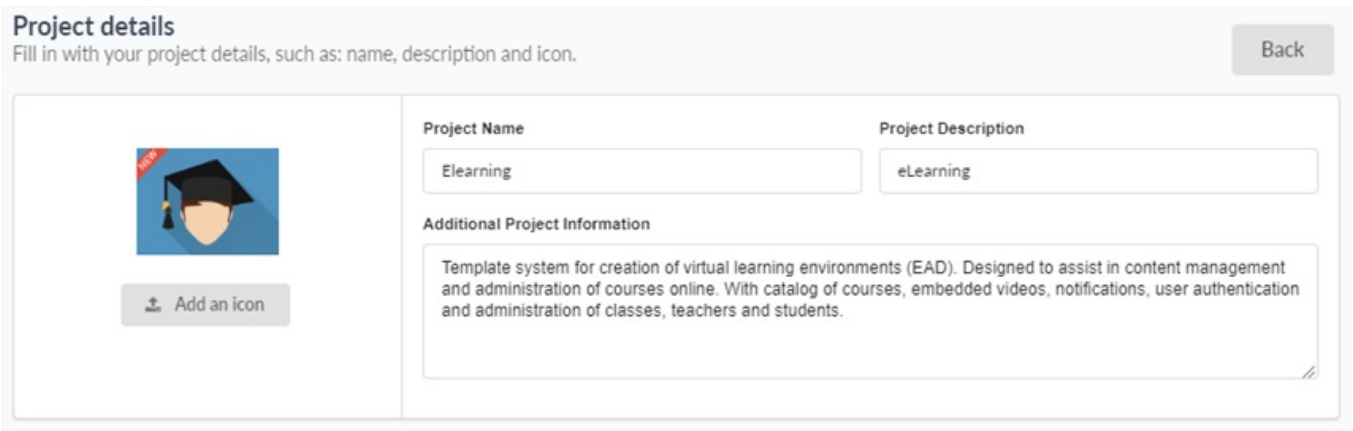
Start

During the first step of the project creation, you can create a **Blank project** or **Import a template** according to the options below.



When you import a project template, Scriptcase will delivery some ready to use applications and also the database tables. Here you see the steps to import one template project, the “Samples.”

First, you need to choose the project and customize all the details. Description and image are optional.



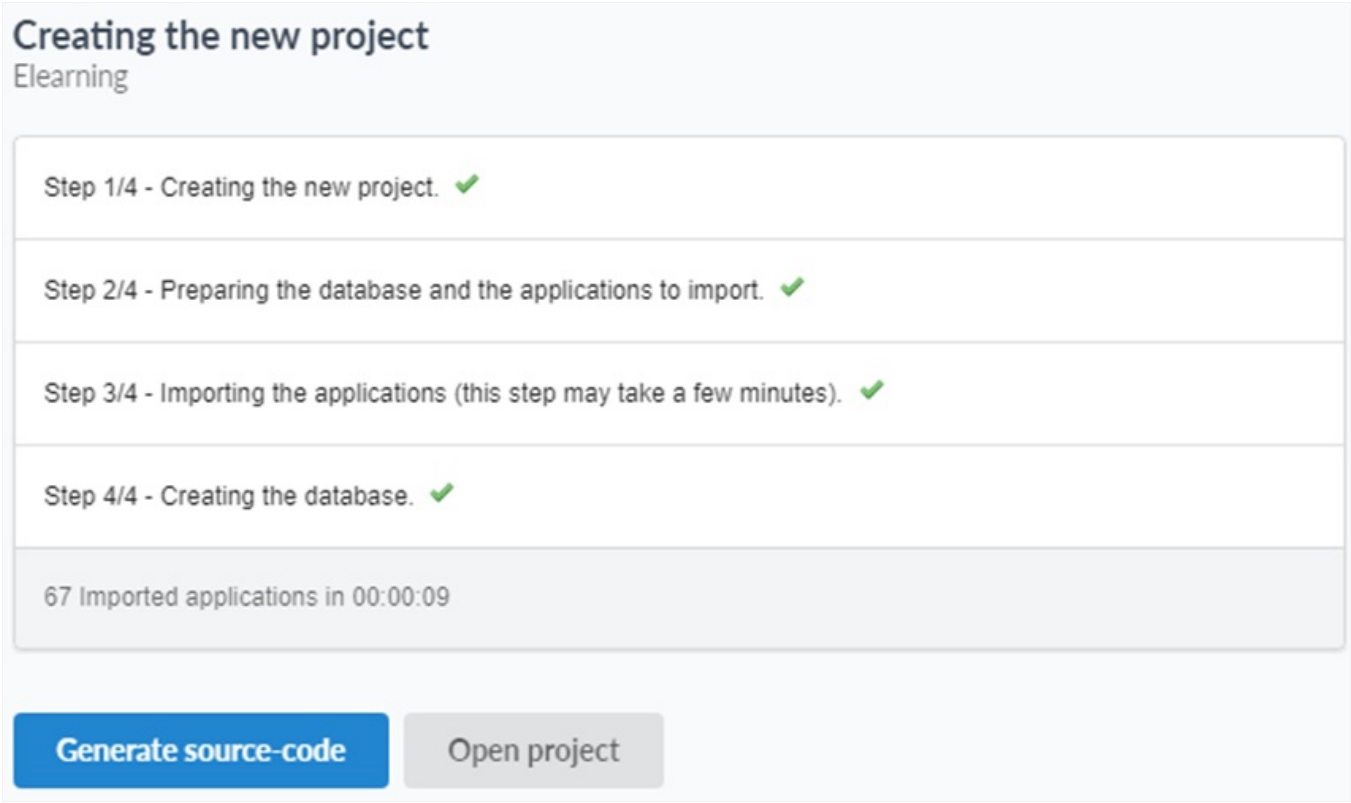
- **Project Icon** - You can select a project icon for display purposes in the project list. This field is not required, and you may change after creation inside project properties.
- **Project Name** - The project name must contain 1-32 alphanumeric characters, without space neither special characters.
- **Project Description** - Here, you give a brief description of the project, not required, also able to inform after the creation of the project.
- **Additional Project information** - Here, you can add some additional information you want for the project. This field is not required, and you may change after the creation of project properties.

Database

Our template projects are available for MySQL, Oracle, SQL Server, PostgreSQL, Access, SQLite, Firebird.
You can choose to use the default connection or to configure a new connection to your database.

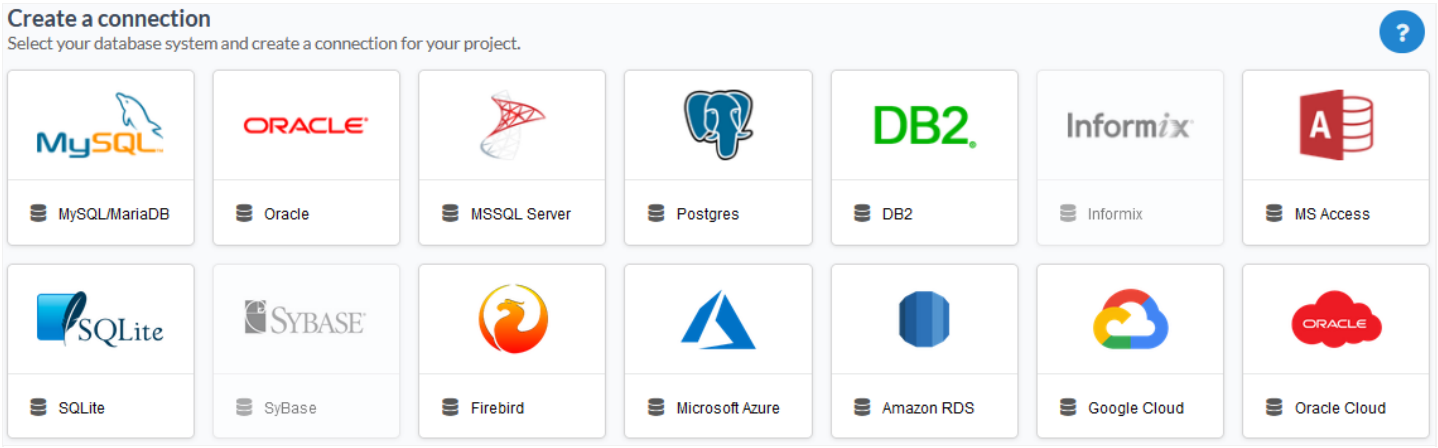
Use default connection (SQLite)

By clicking on the option “Use default connection (SQLite),” Scriptcase undertakes to create a database, with the SQLite database, and insert all the information of tables and records. The database file with the project tables and some dummy records within the Scriptcase directory.



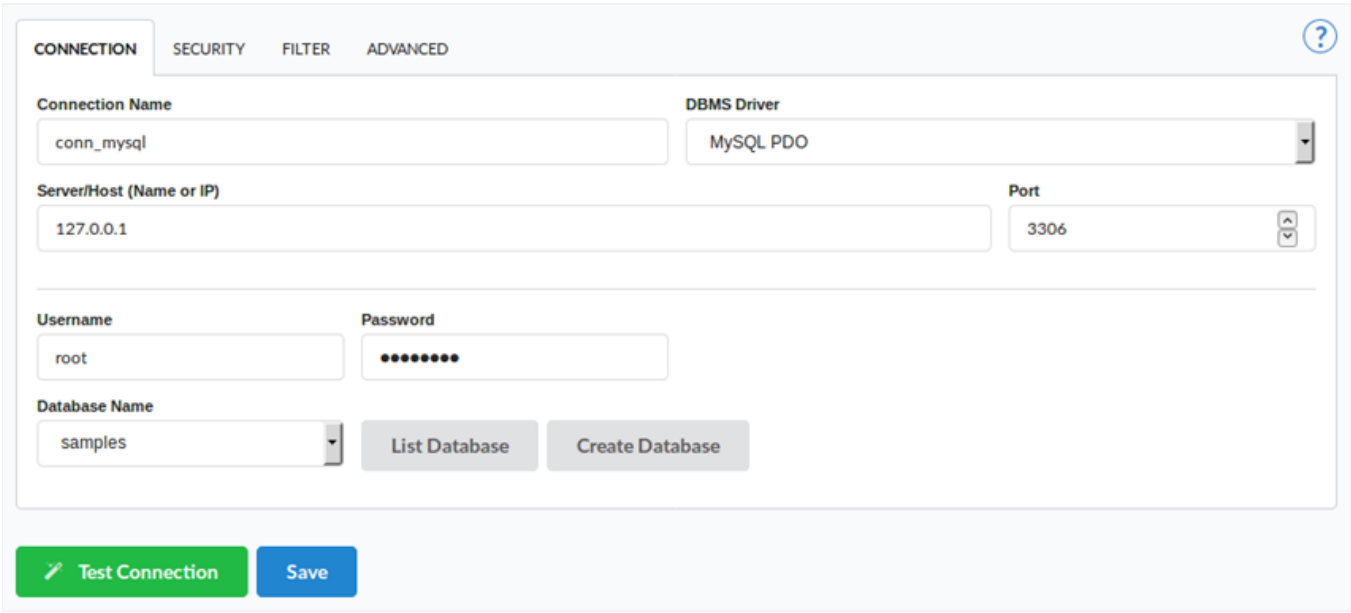
The project will be created and configured with all the essential information. Generate the source code to compile and applications and then “open project” and view all the applications created or “Run” the project to open the initial application

Create a new connection to my database



After selecting the database you want to connect, enter the required data to perform the connection.

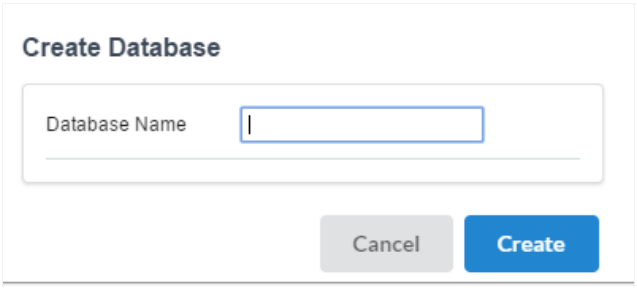
The information needed to create a connection varies according to the selected database. In the example below, we show with MySQL connection. For more details on how to connect your database click here.



- **Connection Name**: Define the name of the connection. You can change this name in the future.
- **DBMS Driver** : Defined as MySQL PDO. It can also be MySQLi.
- **Server/Host (Name or IP)**: Enter the IP of the server where the database is. If the database is on the same machine of Scriptcase, you can use the IP **127.0.0.1** or **localhost**.
- *_Port _*: Define the port of the connection. By default, the port is 3306.
- **User**: Inform the user name that you use to connect to your database.
- **Password**: Inform the password that you use to connect to your database.
- **Database Name**: Click on the “List Database” and select the desired database.
- **Test Connection**: Displays the connection status if it succeeded or not.

If you don’t have a database created, click on the button “Create Database” and the window below appears.

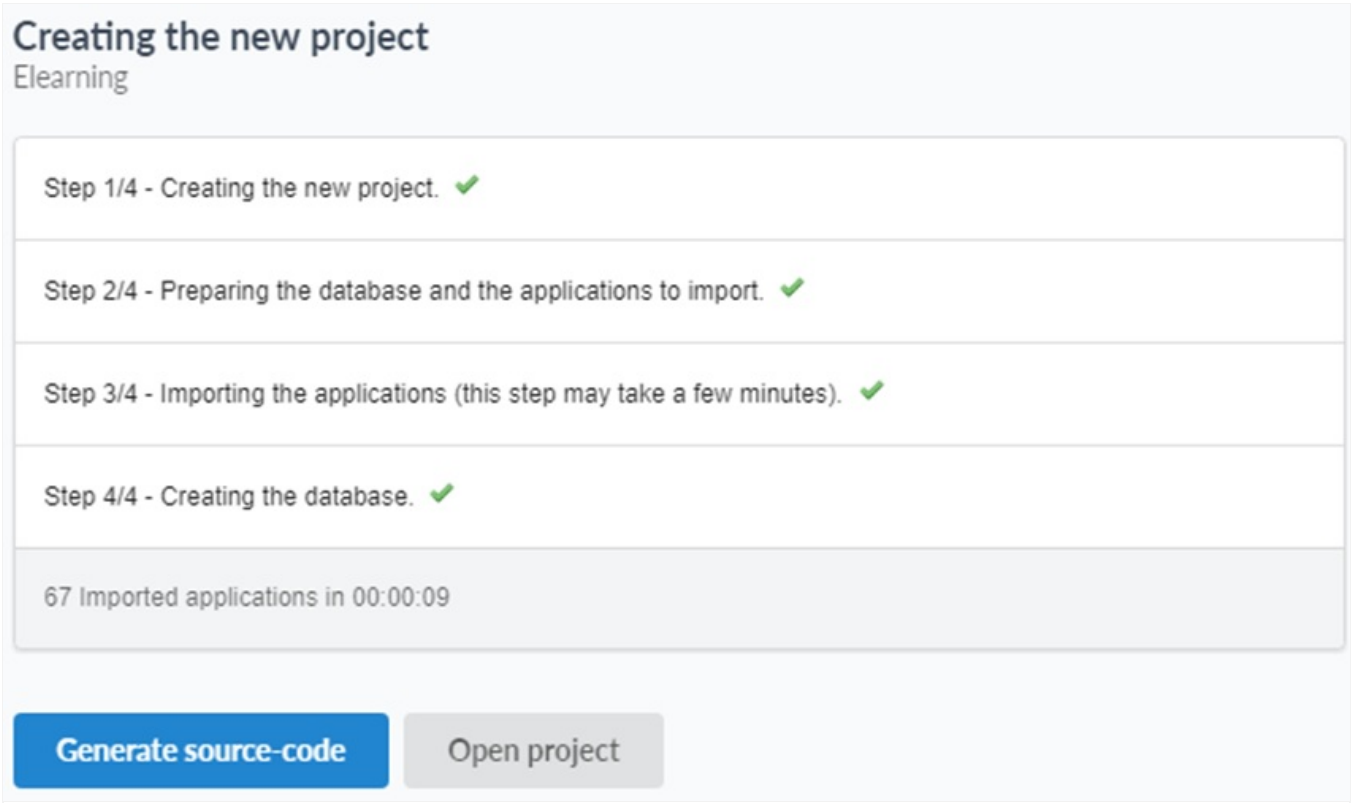
The option of **Create Database** is available for: **MySQL/MariaDB**, **MSSQL Server** and **PostegreSQL**



Enter a name for your database and click “OK.” After that, the database created will be displayed on the database list. To create a database directly from Scriptcase, you need to have the correct permissions within the database system itself, and sometimes this permission is refused, especially when you have the database in another server. In this example, we have the database located on the same server, localhost.

Project Generation

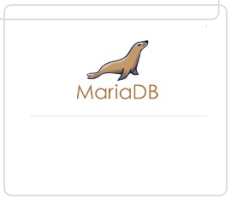
After you finish, Scriptcase undertakes to create all the tables and insert all the information used in applications.

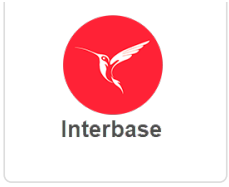


All set! Now the project will be created and configured with all the essential information. Generate the source code to compile the applications and then open project and view all the applications created.

Related Links

ScriptCase Available Connections



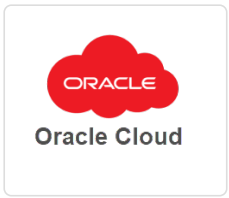

Interbase

Firebird

Microsoft Azure

Amazon RDS

Google Cloud


ORACLE
Oracle Cloud

Progress

Generic
ODBC

Access

In ScriptCase we provide two drivers for the MS ACCESS connection, **MS Access ODBC** and **MS Access ADO.**

You can find out if drivers are available by going to diagnosis. See below how to find your diagnostics and check if the driver is enabled.

- Accessing the top menu **help > Diagnosis**, you can faind it in the interface.

- you can find it too in your browser:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `domain.net/scriptcase/diagnosis.php`

Click on the driver that you need know how to enable.

- Access ODBC
- Access ADO

Connecting with MS Access ODBC

The Scriptcase provides two drivers for connecting to the Access database: **ODBC** and **ADO**. For more information on how to enable the MS Access ADO driver see our documentation.

In Scriptcase, it is also possible to convert tables from Access database files (**.mdb** or **.accdb**) to the databases: **MySQL, PostgreSQL, SQLite and SQL Server**. For more information, see our documentation for Database Convert.

Requirements

Before proceeding with this documentation, check your PHP architecture. For the drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

- **x86** Architecture = 32 Bits
- **x64** Architecture = 64 Bits

  WARNING: The PHP architecture of Scriptcase and MS Access must be equivalent.

In the automatic installation of Scriptcase, the **COM** extension is already enabled in PHP, requiring only the following items below to make the connection with Access:

- The MS Access database (**.mdb** or **.accdb**) must be stored on the same server/machine as Scriptcase with one or more tables created.
- Access Database Engine installed.
- System DSN ODBC created in the Windows ODBC Manager.

Necessary files:

x64

- Access Database Engine **2010**: Click here

x86

- Access Database Engine **2010**: Click here

Installing the Microsoft Access Database Engine 2010 Driver

- First we must download Driver Access Database Engine 2010 according to **your PHP** architecture:

Access Database Engine 2010 x86 Download
Access Database Engine 2010 x64 Download

After download, run the file to install the engine:

1 - Click **“Next”** to continue the installation.

2 - Accept the license terms and agree to continue.

3 - Set the path of the Access Engine installation. You can also use the default path entered by the installer.

4 - Installation complete, click **“OK”** to finish.

Create an ODBC System DSN

See below for all the tutorials for creating an ODBC data source to connect to Scriptcase.

> **NOTE**: The ODBC must be created on the same server where ScriptCase is installed.

1 - Access the **ODBC Data Manager** and select according to architecture. We will select the x64 version:

2 -When you log into ODBC Data Source Manager, select the **DSN** tab and click **Add** to create your connection to your Access database.

3 - After this, you need to select the controller to connect to Access. Select driver: **Microsoft Access Driver (*mdb, *accdb).**

4 - Now you need to define the name of the DSN and select the Access database file.

Data Source Name

Here you will define the name of the DSN to be entered in the Scriptcase connection.

Description

In this item, you can create a description for this DSN by distinguishing it from another one created.

Database

In this section, you must select the database file by clicking the **Select...** button, but you can also create new databases, repair or compress existing files.

5 - After clicking **Select**, you must choose the path to the Access file.

6 - Now just click **“OK”** to confirm the creation of DSN, so it will appear in the list of **DSN** available.

Create a connection in Scriptcase

In this documentation we will show the connection through the ODBC driver by System DSN and in File path to the Access database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

- Select connection **“MS Access”**.

- Select DBMS driver: **MS Access ODBC(Recommended)**

In this access connection controller, we can connect in two ways: using the ODBC DSN and entering the database path in **“Server/Host (Name or IP)”**.

Connect to Scriptcase using the system DSN:

In this type of connection, we must inform the DSN of the system created in step Creating an ODBC data source to make the connection.

Connection name

Define the name of your new connection in Scriptcase.

DBMS driver

Select the access connection controller.

Server/Host (Name or IP)

Enter the system DSN created in Windows ODBC.

Username

Inform the user to connect to the Access database. **This option should only be populated if a user is configured as required when connecting to the Access database.**

Password

Enter the password to connect to the Access database. **This option should be completed only if a password is set as needed when connecting to the Access database.**

Connect to Scriptcase using the file path:

In this type of connection, we must enter the full path to the Access database file to make the connection.

Connection name

Define the name of your new connection in Scriptcase.

DBMS driver

Select the access connection controller.

Server/Host (Name or IP)

Enter the full path to the Access database file.

Username

Inform the user to connect to the Access database. **This option should only be populated if a user is configured as required when connecting to the Access database.**

Password

Enter the password to connect to the Access database. **This option should be completed only if a password is set as needed when connecting to the Access database.**

Filter

We can configure the initial filtering of the information that the database will bring and display.

Show

In this option, we will define if the connection will show the **Tables**, **Views**, **System Tables** or **Procedures** databases.

Searches

In this option we can configure the display of specific tables by user. If the elements are empty, the user informed in authentication will see all the data available.

- **Tables:** We will list the tables that may or may not be displayed.
- **Owner:** We will inform you of the name of the user who has access to the necessary tables or to the complete database.
- **Show:** We will define if the reported elements will be displayed or not in this connection.

Advanced

You can configure more advanced settings for the connection.

Decimal separator

This option allows you to define the decimal separator of numerical values. You can choose between dot (.) and comma (,).

Persistent Connection

Enabling this option will close connections when script execution completes.

Use the schema before the table name

This option causes Scriptcase to use the database schema before the table names in its actions. For example:

dbo.Account

- **dbo**: It is the name of the scheme used.
- **Account**: This is the name of the table that is used soon after.

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection with MS Access ADO

Scriptcase provides two drivers for connecting to the Access database: ADO and ODBC. For more information on how to enable the MS Access ODBC driver, see our documentation.

In Scriptcase, you can also convert tables from Access database files.(**.mdb** or **.accdb**) for databases: **MySQL, PostgreSQL, SQLite and SQL Server**. For more information, see our documentation for database import.

Prerequisites

To create a Scriptcase connection to the Access Database using the ADO connection, you must use the full path to the database file.

In Scriptcase automatic installation, the **COM** extension is already enabled in PHP, and only requires the following elements:

- The PHP architecture of Scriptcase and Access must be equivalent.
- **Microsoft Access Database Engine 2010** client installed.
- The access database (**.mdb** or **.accdb**) must be stored on the same Scriptcase server/machine with one or more tables created.

Checking the PHP Architecture

Before continuing with this documentation, it is important to check your PHP architecture inside **phpinfo()**. If you are using Scriptcase's automatic installer, the PHP architecture will be the same as that of the downloaded installer.

You can access the phpinfo of your Scriptcase through your own path, like:

http://127.0.0.1:8092/scriptcase/info.php

- **x86** Architecture = 32 Bits
- **x64** Architecture = 64 Bits

Installing the Microsoft Access Database Engine 2010 Driver

- First we must download Driver Access Database Engine 2010 according to **your PHP** architecture:

Access Database Engine 2010 x86 Download
Access Database Engine 2010 x64 Download

After download, run the file to install the engine:

1 - Click **"Next"** to continue the installation.

2 - Accept the license terms and agree to continue.

3 - Set the path of the Access Engine installation. You can also use the default path entered by the installer.

4 - Installation complete, click **"OK"** to finish.

Create a connection in Scriptcase

In this documentation we will show the connection to the Access database file.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

- Select connection **“MS Access”**.

- Select DBMS Driver: **MS Access ADO**

Connecting to Scriptcase

In this type of connection, we must enter the full path to the Access database file to make the connection.

Connection name

Define the name of your new connection in Scriptcase.

DBMS Driver

Select the access connection controller.

Server/Host(Name or IP)

Enter the full path to the Access database file.

User

Inform the user to connect to the Access database. **This option should only be completed if a user is configured as required when connecting to the Access database.**

Password

Enter the password to connect to the Access database. **This option should only be completed if a password is set as required when connecting to the Access database.**

Filter

We can configure the initial filtering of the information that the database will bring and display.

Show

In this option we will define if the connection will bring **Tables**, **Views**, **System Tables**, **Procedures**.

Searches

In this option we can configure the display of specific tables by user. If the elements are empty, the user informed in authentication will see all the data available.

- **Tables:** We will list the tables that can be displayed or not.
- **Owner:** We will inform the name of the user who has access to the required tables or to the complete database.

- **Show:** We will define if the reported elements will be shown or not in this connection.

Advanced

You can configure more advanced settings for the connection.

Decimal separator

This option allows you to define the decimal separator of numerical values. You can choose between dot (.) and comma (,).

Persistent Connection

Enabling this option will close connections when script execution completes.

Use the schema before the table name

This option causes Scriptcase to use the database schema before the table names in its actions. For example:

dbo.Account

- **dbo**: It is the name of the scheme used.
- **Account**: This is the name of the table that is used soon after.

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel.

This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

IBM DB2

Scriptcase provides some driver options for connecting to DB2.

You can see if drivers are enabled by going to diagnosis.

Acessando pelo menu do Scriptcase.

After login, in the tool's top menu, access **Help > Diagnosis**.

Accessing directly through the browser

In your browser, without having to login to Scriptcase, you can access the diagnosis.

In case of local installations:

`127.0.0.1:8092/scriptcase/diagnosis.php` ou `localhost:8092/scriptcase/diagnosis.php`

In case of installation on a server.

`domain.com/scriptcase/diagnosis.php`

Click on the driver you want to use, according to your operating system.

Windows

- DB2
- DB2 Native ODBC

Linux

- DB2
- DB2 Native ODBC

Linux

- DB2
- DB2 Native ODBC

Connecting with DB2

In Scriptcase, we have the following drivers available for connection to DB2: **DB2 and DB2 Native ODBC.** If you are using your own, pre-configured environment, DB2 extensions **must be manually enabled** in PHP.

Automatic Installation

> **If you installed the tool through the automatic installer, and did not install the driver at the time of installation. You must download the DB2 driver installer according to your architecture and install it. The installation will install the client and enable the necessary extensions.**

To check the architecture, just access http://127.0.0.1:8092/scriptcase/info.php

x64

- scriptcase_database_drivers_v9.9-x64-php8.1: Download

x86

- scriptcase_database_drivers_v9.9-x64-php8.1: Download

After installing the drivers, go to how to configure the DSN connection by clicking here

Manual Installation

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for drivers to be enabled correctly, **files must be downloaded according to the used architecture.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Listed below are the files required for DB2 drivers to be enabled.

Required files:

x64

- IBM Data Server Runtime Client **11.5(x64)**: Click here

x86

- IBM Data Server Runtime Client **11.5(x86)**: Click here

Other Files

- PHP extensions **(PHP_PDO_IBM and PHP_IBM_DB2)**: Click here

Configuring DB2 on Windows

Before installing the IBM Data Server Runtime Client, you need to enable the extensions in PHP for Scriptcase to recognize the DB2 connection drivers. Follow them steps below to correctly enable all drivers in Scriptcase.

1 - Extract the .zip file and copy the downloaded **php_pdo_ibm** and **php_ibm_db2** .dll extension files previously to your PHP ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the DB2 extensions **php_pdo_ibm** and **php_ibm_db2**. See the example below:

```
extension=ibm_db2
extension=pdo_ibm
```

3 - After downloading, follow the installation wizard by clicking on next, keeping the default values of the installer.

4 - Accept the terms and proceed with the installation.

5 - Click Next, to keep the default values of the installer.

6 - Click Install, to start the installation process.

7 - Click Finish to complete the installation.

8 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache or apachesc9php81 service and **Restart** the service.

Creating the DSN for connection

To connect to the **DB2 PDO ODBC** Driver, you need to manually create a DataSource Name in the DB2 files. Follow the steps described below to perform this configuration.

> **ATTENTION**: The ODBC must be created on the same server where Scriptcase is installed.

1 - Access your **ODBC Data Source Manager** and select according to your architecture. We will select the x64 version:

2 - When entering the ODBC Data Sources Administrator, select the **DSN System** tab and click **Add** to create your connection to your Access Database.

3 - After that, it is necessary to select the Driver to connect to DB2. Select the Driver: **IBM DB2 ODBC DRIVER.**

4 - Now, you need to define the DSN name and the Database alias.

- **Data Source Name:** Enter the name that the DSN will have so that you can use it in Scriptcase.
- **Database alias:** Include and enter the name of the database you will connect to.
- **Description:** Add a description for the DSN.

4.1 - After entering the name and description, click the button. After informing the name and description, click the **Include** button.

- **User ID:** Inform the user to authenticate with the DB2 database.
- **Password:** Enter the corresponding password to authenticate with the informed user.
- **Save Password:** By checking this option, the password is stored for future authentications and connection tests.

4.2 - Now, to configure the database, click on the **TCP/IP** tab.

- **Database Name:** Inform the database that you will connect to.
- **Database alias:** Enter the database alias, if the database has one. If not, use the same name as the database.
- **Hostname:** Enter the IP or domain of the server where the DB2 database is installed.
- **Port Number:** Enter the port for connection. By default, DB2 banks use port `50000`.

To finish, click on **OK**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the DB2 database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **DB2** connection.

Connection

Enter the parameters for connecting to your database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Driver for connection. In this example, we use Driver **DB2**.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
- **EX:** `dominiodoserver.com` or `192.168.254.170`
- **Port**: Enter the port to connect to DB2. By default, the defined port is **50000**.
- **Database Name**: Enter the name of the database you will connect to.
- **EX:** `SAMPLE`
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database .**
- **Username**: Inform the user to authenticate the connection to your database.

- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Here are the security options available for a DB2 connection and their descriptions:

Security

Defines the security mechanism used for the connection. It can include options like SSL, Kerberos, or other authentication protocols.

sslservercertificate

Specifies the server's SSL certificate file used to establish a secure connection. This helps verify the identity of the database server.

sslclientkeystoredb

Refers to the client-side keystore database file that contains SSL certificates and keys necessary for encrypted communication with the DB2 server.

sslclientkeystash

The keystore stash file that contains the encrypted password for the keystore database, allowing the client to access SSL keys securely without manual intervention.

Authentication

Defines the authentication method used for connecting to the DB2 server. It can include methods such as SERVER, CLIENT, or GSSAPI.

sslclientlabel

Specifies the label of the client certificate stored in the keystore database. This helps identify which certificate should be used when multiple certificates exist in the keystore.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with DB2 Native ODBC

In Scriptcase, we have the following drivers available for connection to DB2: **DB2 and DB2 Native ODBC.** If you are using your own, pre-configured environment, DB2 extensions **must be manually enabled** in PHP.

Automatic Installation

> **If you installed the tool through the automatic installer, and did not install the driver at the time of installation. You must download the DB2 driver installer according to your architecture and install it. The installation will install the client and enable the necessary extensions.**

To check the architecture, just access http://127.0.0.1:8092/scriptcase/info.php

x64

- scriptcase_database_drivers_v9.9-x64-php8.1: Download

x86

- scriptcase_database_drivers_v9.9-x86-php8.1: Download

Manual Installation

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for drivers to be enabled correctly, **files must be downloaded according to the used architecture.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Listed below are the files required for DB2 drivers to be enabled.

Required files:

x64

- IBM Data Server Runtime Client **11.5(x64)**: Click here

x86

- IBM Data Server Runtime Client **11.5(x86)**: Click here

Other Files

- PHP extensions **(PHP_PDO_IBM and PHP_IBM_DB2)**: Click here

Configuring DB2 Native ODBC on Windows

Before installing the IBM Data Server Runtime Client, you need to enable the extensions in PHP for Scriptcase to recognize the DB2 connection drivers. Follow them steps below to correctly enable all drivers in Scriptcase.

1 - Extract the .zip file and copy the downloaded **php_pdo_ibm** and **php_ibm_db2** .dll extension files previously to your PHP ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the DB2 extensions **php_pdo_ibm** and

php_ibm_db2. See the example below:

```
extension=ibm_db2
extension=pdo_ibm
```

3 - After downloading, follow the installation wizard by clicking on next, keeping the default values of the installer.

4 - Accept the terms and proceed with the installation.

5 - Click Next, to keep the default values of the installer.

6 - Click Install, to start the installation process.

7 - Click Finish to complete the installation.

8 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache or apachesc9php81 service and **Restart** the service.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the DB2 database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **DB2** connection.

Connection

Enter the parameters for connecting to your database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Driver for connection. In this example, we use Driver **DB2 Native ODBC**.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
- **EX:** dominiodoserver.com or 192.168.254.170

- **Port**: Enter the port to connect to DB2. By default, the defined port is **50000**.
- **Database Name**: Enter the name of the database you will connect to.
- **EX:** SAMPLE
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database .**
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Here are the security options available for a DB2 connection and their descriptions:

Security

Defines the security mechanism used for the connection. It can include options like SSL, Kerberos, or other authentication protocols.

sslservercertificate

Specifies the server’s SSL certificate file used to establish a secure connection. This helps verify the identity of the database server.

sslclientkeystoredb

Refers to the client-side keystore database file that contains SSL certificates and keys necessary for encrypted communication with the DB2 server.

sslclientkeystash

The keystore stash file that contains the encrypted password for the keystore database, allowing the client to access SSL keys securely without manual intervention.

Authentication

Defines the authentication method used for connecting to the DB2 server. It can include methods such as SERVER, CLIENT, or GSSAPI.

sslclientlabel

Specifies the label of the client certificate stored in the keystore database. This helps identify which certificate should be used when multiple certificates exist in the keystore.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to PDO DB2

At Scriptcase, we have the following drivers available for connection to DB2: **DB2 and DB2 Native ODBC.** If you are using your own, pre-configured environment , DB2 extensions must be enabled manually in PHP.

Automatic Installation

> **If you installed the tool through the automatic installer, and did not install the driver at the time of installation. You must download the DB2 driver installer and install it. The installation will install the client and enable the necessary extensions.**

- scriptcase_database_drivers_linux_amd64_v9.9.run : Download

Running the installer

Enter the folder where the installer is located and grant the installer permission

```
sudo chmod +x scriptcase_database_drivers_linux_amd64_v9.9.run
```

Run the installer

```
./scriptcase_database_drivers_linux_amd64_v9.9.run
```

After installing the drivers, go to how to configure the **db2dsdriver.cfg** clicking here

Manual Installation

Prequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **files must be downloaded according to the architecture used .**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  CAUTION: Download the dsdriver according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Required files:

x64

- IBM DsDriver **Linux(x64)**: click here

x86

- IBM DsDriver **Linux(x86)**: click here

Follow them steps below to correctly enable all drivers in Scriptcase.

1 - Install the dependencies below:

UBUNTU\DEBIAN	CENTOS\RHEL
- sudo apt-get update	- sudo yum update
- sudo apt-get install libaio1	- sudo yum install libaio-devel
- sudo apt-get install libncurses5	- sudo yum install libaio
- sudo apt-get install alien	- sudo yum install glibc
- sudo apt-get install gcc-multilib g++-multilib	- sudo yum install compat-libstdc++-33
- sudo apt-get install libpam0g	- sudo yum install glibc-devel
- sudo apt-get install ksh	- sudo yum install libstdc++

UBUNTU\DEBIAN	CENTOS\RHEL
- sudo apt-get install unixodbc-dev unixodbc	- sudo yum install libstdc++
	- sudo yum install pam
	- sudo yum install ncurses-devel
	- sudo yum install unixODBC
	- sudo wget ftp://195.220.108.108/linux/centos/6.7/os/i386/Packages/ksh-20120801-28.el6.i686.rpm
	- sudo yum install ksh-20120801-28.el6.i686.rpm
	- sudo yum install nano

2 - Install the IBM DB2 DsDriver:

ARCHITECTURE x86	ARCHITECTURE x64
- sudo mkdir -p /opt/IBM/Db2	- sudo mkdir -p /opt/IBM/Db2
- sudo cp sudo cp x86_v10.5fp8_linuxia32_dsdriver.tar.gz /opt/IBM/Db2	- sudo cp x64_v10.5fp8_linuxx64_dsdriver.tar.gz /opt/IBM/Db2
- sudo tar -zxf /opt/IBM/Db2/x86_v10.5fp8_linuxia32_dsdriver.tar.gz	- sudo tar -zxf /opt/IBM/Db2/x64_v10.5fp8_linuxx64_dsdriver.tar.gz
- sudo ksh /opt/IBM/Db2/dsdriver/installDSDriver	- sudo ksh /opt/IBM/Db2/dsdriver/installDSDriver

3 - Restart the Apache service:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Configuring the db2dsdriver.cfg file

Before creating the connection in the Scriptcase interface, you need to configure the **db2dsdriver.cfg** file

1 : Give recursive permission on the file directory if you do not have permission to manipulate the file.

Automatic Installation	Manual Installation
- sudo chmod -R 755 /opt/Scriptcase/v9-php81/components/drivers/IBM/Db2/dsdriver/cfg/	- sudo chmod -R 755 /opt/IBM/Db2/dsdriver/cfg/

2 : Open the db2dsdriver.cfg file and configure the file as in the structure below

```
<configuration>
<dsncollection>
   <dsn alias="db2" name="db2" host="hotel53.torolab.ibm.com" port="5000" ldap=1/>
</dsncollection>
   <databases>
      <database name="SAMPLE" host="hotel53.torolab.ibm.com" port="5000">
      </database>
   </databases>
</configuration>
```

In automatic installation, you must first rename the **db2dsdriver.cfg.sample** file to **db2dsdriver.cfg**

In manual installation, if you don’t find the file, you can create it

After configuring the file, restart Apache and create the connection.

Automatic Installation	Manual Installation
UBUNTU: sudo service apachesc9php81 restart	sudo service apache2 restart
CENTOS/RHEL: sudo service apachesc9php81 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the DB2 database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **DB2** connection.

Connection

Enter the parameters for connecting to your database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Driver for connection. In this example, we use the **DB2 ODBC Genérico** driver.
- **Server/Host (Name or IP)**: Enter the System DSN created in the ODBC Data Source.
- **EX:** db2
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database .**
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Here are the security options available for a DB2 connection and their descriptions:

Security

Defines the security mechanism used for the connection. It can include options like SSL, Kerberos, or other authentication protocols.

sslservercertificate

Specifies the server’s SSL certificate file used to establish a secure connection. This helps verify the identity of the database server.

sslclientkeystoredb

Refers to the client-side keystore database file that contains SSL certificates and keys necessary for encrypted communication with the DB2 server.

sslclientkeystash

The keystore stash file that contains the encrypted password for the keystore database, allowing the client to access SSL keys securely without manual intervention.

Authentication

Defines the authentication method used for connecting to the DB2 server. It can include methods such as SERVER, CLIENT, or GSSAPI.

sslclientlabel

Specifies the label of the client certificate stored in the keystore database. This helps identify which certificate should be used when multiple certificates exist in the keystore.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed

before the table names.
- **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to DB2 Native ODBC

At Scriptcase, we have the following drivers available for connection to DB2: **DB2 and DB2 Native ODBC.** If you are using your own, pre-configured environment , DB2 extensions must be enabled manually in PHP.

Automatic Installation

> **If you installed the tool through the automatic installer, and did not install the driver at the time of installation. You must download the DB2 driver installer and install it. The installation will install the client and enable the necessary extensions.**

- scriptcase_database_drivers_linux_amd64_v9.9.run : Download

Running the installer

Enter the folder where the installer is located and grant the installer permission

```
sudo chmod +x scriptcase_database_drivers_linux_amd64_v9.9.run
```

Run the installer

```
./scriptcase_database_drivers_linux_amd64_v9.9.run
```

After the installation, restart Scriptcae apache

```
sudo service apachesc9php81 restart
```

Manual Installation

Prequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **files must be downloaded according to the architecture used .**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  CAUTION: Download the IBM DsDriver client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Required files:

x64

- IBM DsDriver **Linux(x64)**: click here

x86

- IBM DsDriver **Linux(x86)**: click here

Follow the steps below to correctly enable all drivers in Scriptcase.

1 - Install the dependencies below:

UBUNTU\DEBIAN	CENTOS\RHEL
- sudo apt-get update	- sudo yum update
- sudo apt-get install libaio1	- sudo yum install libaio-devel
- sudo apt-get install libncurses5	- sudo yum install libaio
- sudo apt-get install alien	- sudo yum install glibc

UBUNTU\DEBIAN	CENTOS\RHEL
- sudo apt-get install gcc-multilib g++-multilib	- sudo yum install compat-libstdc++-33
- sudo apt-get install libpam0g	- sudo yum install glibc-devel
- sudo apt-get install ksh	- sudo yum install libstdc++
- sudo apt-get install unixodbc-dev unixodbc	- sudo yum install libstdc++
	- sudo yum install pam
	- sudo yum install ncurses-devel
	- sudo yum install unixODBC
	- sudo wget ftp://195.220.108.108/linux/centos/6.7/os/i386/Packages/ksh-20120801-28.el6.i686.rpm
	- sudo yum install ksh-20120801-28.el6.i686.rpm
	- sudo yum install nano

2 - Install the IBM DB2 DsDriver:

ARCHITECTURE x86	ARCHITECTURE x64
- sudo mkdir -p /opt/IBM/Db2	- sudo mkdir -p /opt/IBM/Db2
- sudo cp sudo cp x86_v10.5fp8_linuxia32_dsdriver.tar.gz /opt/IBM/Db2	- sudo cp x64_v10.5fp8_linuxx64_dsdriver.tar.gz /opt/IBM/Db2
- sudo tar -zxf /opt/IBM/Db2/x86_v10.5fp8_linuxia32_dsdriver.tar.gz	- sudo tar -zxf /opt/IBM/Db2/x64_v10.5fp8_linuxx64_dsdriver.tar.gz
- sudo ksh /opt/IBM/Db2/dsdriver/installDSDriver	- sudo ksh /opt/IBM/Db2/dsdriver/installDSDriver

3 - Restart the Apache service:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the DB2 database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **DB2** connection.

Connection

Enter the parameters for connecting to your database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Driver for connection. In this example, we use Driver **DB2 Native ODBC**.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
- **EX:** dominiodoserver.com or 192.168.254.170

- **Port**: Enter the port to connect to DB2. By default, the defined port is **50000**.
- **Database Name**: Enter the name of the database you will connect to.
- **EX:** SAMPLE
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database .**
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Here are the security options available for a DB2 connection and their descriptions:

Security

Defines the security mechanism used for the connection. It can include options like SSL, Kerberos, or other authentication protocols.

sslservercertificate

Specifies the server’s SSL certificate file used to establish a secure connection. This helps verify the identity of the database server.

sslclientkeystoredb

Refers to the client-side keystore database file that contains SSL certificates and keys necessary for encrypted communication with the DB2 server.

sslclientkeystash

The keystore stash file that contains the encrypted password for the keystore database, allowing the client to access SSL keys securely without manual intervention.

Authentication

Defines the authentication method used for connecting to the DB2 server. It can include methods such as SERVER, CLIENT, or GSSAPI.

sslclientlabel

Specifies the label of the client certificate stored in the keystore database. This helps identify which certificate should be used when multiple certificates exist in the keystore.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection DB2

At Scriptcase, we have the following drivers available for connection to DB2: **DB2 and DB2 Native ODBC.** If you are using your own, pre-configured environment , DB2 extensions must be enabled manually in PHP.

> **Note:** The steps described in this documentation are only valid for Mac devices that have CPU with **x64** architecture

Installing IBM DSDRIVER

Download the IBM DSDRIVER installer

- IBM DsDriver: Click here

Follow the steps below to install correctly.

1 - Double-click the **v11.1.1fp1_macos_dsdriver.dmg** file

A new Finder window will open with the contents of the disk image.

2- Run the installation script:

- In the Finder window, locate the installDSDriver.sh. file
- Double-click the installDSDriver.sh file to start the installation.
- The driver package will be installed in the default location: /Applications/dsdriver.

Setting the environment variables

For PHP to recognize the Db2 driver, set the following environment variables:

Open the ~/.bash_profile or ~/.zshrc file (depending on the shell you are using) in a text editor and add the following line

```
export IBM_DB_HOME=/Applications/dsdriver
export DYLD_LIBRARY_PATH=$IBM_DB_HOME/lib
export PATH=$IBM_DB_HOME/bin:$PATH
```

Save the file and run the following command to apply the changes:

source ~/.bash_profile or source ~/.zshrc

Setting the db2dsdriver.cfg file

Before creating the connection in the Scriptcase interface, you need to set up the file **db2dsdriver.cfg**

> The file may be named *db2dsdriver.cfg.sample.* Just create a copy of this file and rename it to **db2dsdriver.cfg**

If you can't find the file, you can also create it manually.

1 : Open the **db2dsdriver.cfg** file located in **/Applications/dsdriver/cfg/** and configure the file as shown in the structure below and then save.

```
<configuration>
<dsncollection>
   <dsn alias="db2" name="db2" host="hotel53.torolab.ibm.com" port="5000" ldap=1/>
</dsncollection>
   <databases>
      <database name="SAMPLE" host="hotel53.torolab.ibm.com" port="5000">
      </database>
   </databases>
</configuration>
```

After configuring the file, restart Apache and create the connection.

Automatic Installation **Manual Installation**

Automatic Installation	Manual Installation
sudo service apachesc9php81 restart	sudo service apache2 restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the DB2 database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **DB2** connection.

Connection

Enter the parameters for connecting to your database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Driver for connection. In this example, we use Driver **DB2**.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
- **EX:** dominiodoserver.com or 192.168.254.170
- **Port**: Enter the port to connect to DB2. By default, the defined port is **50000**.
- **Database Name**: Enter the name of the database you will connect to.
- **EX:** SAMPLE
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database .**
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Here are the security options available for a DB2 connection and their descriptions:

Security

Defines the security mechanism used for the connection. It can include options like SSL, Kerberos, or other authentication protocols.

sslservercertificate

Specifies the server's SSL certificate file used to establish a secure connection. This helps verify the identity of the database server.

sslclientkeystoredb

Refers to the client-side keystore database file that contains SSL certificates and keys necessary for encrypted communication with the DB2 server.

sslclientkeystash

The keystore stash file that contains the encrypted password for the keystore database, allowing the client to access SSL keys securely without manual intervention.

Authentication

Defines the authentication method used for connecting to the DB2 server. It can include methods such as SERVER, CLIENT, or GSSAPI.

sslclientlabel

Specifies the label of the client certificate stored in the keystore database. This helps identify which certificate should be used when multiple certificates exist in the keystore.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to DB2 Native ODBC

At Scriptcase, we have the following drivers available for connection to DB2: **DB2 and DB2 Native ODBC.** If you are using your own, pre-configured environment , DB2 extensions must be enabled manually in PHP.

> **Note:** The steps described in this documentation are only valid for Mac devices that have CPU with **x64** architecture

Installing IBM DSDRIVER

Download the IBM DSDRIVER installer

- IBM DsDriver: Click here

Follow the steps below to install correctly.

1 - Double-click the **v11.1.1fp1_macos_dsdriver.dmg** file

A new Finder window will open with the contents of the disk image.

2- Run the installation script:

- In the Finder window, locate the installDSDriver.sh. file
- Double-click the installDSDriver.sh file to start the installation.
- The driver package will be installed in the default location: /Applications/dsdriver.

Setting the environment variables

For PHP to recognize the Db2 driver, set the following environment variables:

Open the ~/.bash_profile or ~/.zshrc file (depending on the shell you are using) in a text editor and add the following line

```
export IBM_DB_HOME=/Applications/dsdriver
export DYLD_LIBRARY_PATH=$IBM_DB_HOME/lib
export PATH=$IBM_DB_HOME/bin:$PATH
```

Save the file and run the following command to apply the changes:

source ~/.bash_profile or source ~/.zshrc

Configuring the ODBC Driver Manager

1 - Open ODBC manager, and in **Driver tab**, add **DB2 native driver**. Specify ‘DB2 Native Driver’ in the **‘Driver Name’** slot and **‘/Applications/dsdriver/lib/libdb2.dylib’** in the ‘Driver File’ slot - no other information is required here.

2 - In the **‘System DSN’** tab of ODBC Manager create a DB2 DSN

- In the **“Choose a Driver”** slot specify the **‘DB2 Native Driver’** you defined above
- Specify a DSN in the **‘Data Source Name (DSN)** slot, eg. ‘db2’ and click on ‘Add’ to create the system DSN.

Setting the db2dsdriver.cfg file

Before creating the connection in the Scriptcase interface, you need to set up the file **db2dsdriver.cfg**

> The file may be named *db2dsdriver.cfg.sample.* Just create a copy of this file and rename it to **db2dsdriver.cfg**

If you can’t find the file, you can also create it manually.

1 : Open the **db2dsdriver.cfg** file located in **/Applications/dsdriver/cfg/** and configure the file as shown in the structure below and then save.

```
<configuration>
```

```
<dsncollection>
   <dsn alias="db2" name="db2" host="hotel53.torolab.ibm.com" port="5000" ldap=1/>
</dsncollection>
   <databases>
      <database name="SAMPLE" host="hotel53.torolab.ibm.com" port="5000">
      </database>
   </databases>
</configuration>
```

> The .cfg file must correspond to the configuration made in the ODBC manager, i.e., the alias and name of the DSN must be the same as that specified in step 2 of the **ODBC Driver Manager** configuration.

After configuring the file, restart Apache and create the connection.

Automatic Installation	**Manual Installation**
sudo service apachesc9php81 restart	sudo service apache2 restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the DB2 database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **DB2** connection.

Connection

Enter the parameters for connecting to your database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Driver for connection. In this example, we use Driver **DB2 Native ODBC**.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
- **EX:** dominiodoserver.com or 192.168.254.170
- **Port**: Enter the port to connect to DB2. By default, the defined port is **50000**.
- **Database Name**: Enter the name of the database you will connect to.
- **EX:** SAMPLE
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database .**
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Here are the security options available for a DB2 connection and their descriptions:

Security

Defines the security mechanism used for the connection. It can include options like SSL, Kerberos, or other authentication protocols.

sslservercertificate

Specifies the server’s SSL certificate file used to establish a secure connection. This helps verify the identity of the database server.

sslclientkeystoredb

Refers to the client-side keystore database file that contains SSL certificates and keys necessary for encrypted communication with the DB2 server.

sslclientkeystash

The keystore stash file that contains the encrypted password for the keystore database, allowing the client to access SSL keys securely without manual intervention.

Authentication

Defines the authentication method used for connecting to the DB2 server. It can include methods such as SERVER, CLIENT, or GSSAPI.

sslclientlabel

Specifies the label of the client certificate stored in the keystore database. This helps identify which certificate should be used when multiple certificates exist in the keystore.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a

PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Firebird with Scriptcase

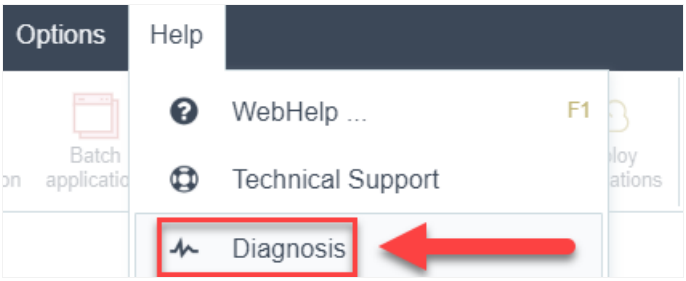
Scriptcase, installed in environments with PHP 8.1, provides only the **Firebird PDO** driver to make the connection with Firebird.

For environments with PHP 7.3 the Firebird driver is also available. See the articles according to your operating system: Firebird in Scriptcase with PHP 7.3 - Windows Firebird in Scriptcase with PHP 7.3 - Linux Firebird in Scriptcase with PHP 7.3 - MacOS

In Scriptcase, we have two drivers to connect with Firebird. You must choose the most suitable for your environment. It is recommended to use **Firebird PDO** which is more faster, however it is up to you to choose the driver. **If you are using your own pre-configured environment, Firebird extensions must be manually enabled in PHP**.

You can check your enabled drivers by accessing your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.



- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

If the desired driver is already enabled and you just want to know how to connect with Scriptcase, click on the link below according to the driver that will be used.

- Windows - Firebird PDO
- Linux - Firebird PDO
- macOS - Firebird PDO

Connecting with Firebird PDO

In Scriptcase, we have the following drivers available for Firebird connections: **Firebird PDO.** If you are using your own pre-configured environment, the Firebird extensions must be enabled manually in PHP.

Prerequisites

If you are using a manual installation on Windows, you will need to enable the Firebird extensions in the **php.ini** file. See below for how to do this.

1 - In the php.ini file located under C:\php, uncomment the line for the Firebird extension **pdo_firebird** by removing the ; from the start of the line. See the example below:

2 - Restart the Apache service using the **Task Manager**.

- Open Task Manager and click on the **Services** tab.
- Look for the Apache2.4 or ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a connection in Scriptcase

Here's how to create a connection in your Scriptcase project using the Firebird database.

1 - Access any project from your Scriptcase.

2 - Click on the **New connection** icon to create a connection.

or go to the **Database > New Connection** menu.

After that a new page will appear with all the database connections.

3 - Select the **Firebird** connection.

Connection

Enter the parameters for connecting to your Firebird database as follows:

- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Firebird Driver used to connect. In this example we use the **Firebird PDO** Driver.
- **Server/Host (Name or IP):** Enter the IP or domain of the server where the database is installed.
  - **Ex:** serverdomain.com or 192.168.254.170.
- **Port**: Enter the numeric port for your database server.
  - **Ex:** 3050
- **Database Name**: Enter the database created for use by the entered user.
- **Username**: Enter the user to authenticate the connection to your Firebird database.
- **Password:** Enter the password to complete the authentication process.
- **Test Connection**: Click this button to get a response from Scriptcase request to know if the entered parameters are correct.

Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.

Show

Allows the Firebird connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option will display your database views.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables are displayed. The setting can contain a PREFIX% or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

- **Owner:** Inform the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above.**
- **Show:** Choose whether tables are displayed for the owner entered.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to connection-specific settings. Changes made in this section impact the data display and performance of the application.

Charset

Select the encoding used in your database.

Role

Defines the user role for the database session. Roles are used to manage permissions and access control within Firebird.

You can set more than one role for the user separating them by using the “;”

Dialect

Determines the SQL dialect used in the connection. Firebird supports different dialects (1, 2, or 3), which affect SQL syntax and behavior.

If the option in the database is set to 2, it doesn’t matter what option you choose in this field.

- **Options**:
  - **Empty** - Gets the default from the database.
  - **1** - Old version of Firebird (1.5) > Does not support timestamp, Numeric or Decimal
  - **2** - Version with timestamp support
  - **3** - Version - Newer versions (Case sensitive and compatible with Firebird 2.5, 3 and 4)

Decimal Separator

Select the separator type for decimal records, between comma and period.

- **By default, period . is selected as the separator.**

Persistent connection

Set whether connections will be closed after your scripts run in Scriptcase applications.

- **By default, Scriptcase disables this option.**

Use schema before table name

Set whether the database schema is displayed before table names.

- **By default, Scriptcase enables this option.**
  - **Ex:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Firebird PDO

In Scriptcase, we have the following driver available for Firebird connection: **Firebird PDO.** If you are using your own pre-configured environment, the Firebird extensions must be enabled manually in PHP.

Prerequisites

If you are using a manual installation on Linux, you will need to install the Firebird php extension. See below for how to do this.

1 - Log into your **Linux** terminal and type this line below according to your operating system to install the Firebird PDO driver.

Ubuntu	CentOS
sudo apt-get install php8.1-interbase	sudo yum install php-interbase

2 - Check that the Firebird driver is enabled in your Scriptcase diagnostics. Here is how to locate your diagnostics and verify that the driver is enabled.

- By accessing the top menu **Help > Diagnostics**, you can find it easily through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a connection in Scriptcase

Here's how to create a connection in your Scriptcase project using the Firebird database.

1 - Access any project from your Scriptcase.

2 - Click on the **New connection** icon to create a connection.

or go to the **Database > New Connection** menu.

After that a new page will appear with all the database connections.

3 - Select the **Firebird** connection.

Connection

Enter the parameters for connecting to your Firebird database as follows:

- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix**conn** along with the database name.
- **DBMS Driver**: Select the Firebird Driver used to connect. In this example we use the**Firebird PDO** Driver.
- **Server/Host (Name or IP):** Enter the IP or domain of the server where the database is installed.
  - **Ex:** serverdomain.com or 192.168.254.170.
- **Port**: Enter the numeric port for your database server.
  - **Ex:** 3050
- **Database Name**: Enter the database created for use by the entered user.
- **Username**: Enter the user to authenticate the connection to your Firebird database.

- **Password:** Enter the password to complete the authentication process.
- **Test Connection**: Click this button to get a response from Scriptcase request to know if the entered parameters are correct.

Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.

Show

Allows the Firebird connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option will display your database views.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables are displayed. The setting can contain a PREFIX% or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

- **Owner:** Inform the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above.**
- **Show:** Choose whether tables are displayed for the owner entered.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to connection-specific settings. Changes made in this section impact the data display and performance of the application.

Charset

Select the encoding used in your database.

Role

Defines the user role for the database session. Roles are used to manage permissions and access control

within Firebird.

You can set more than one role for the user separating them by using the “;”

Dialect

Determines the SQL dialect used in the connection. Firebird supports different dialects (1, 2, or 3), which affect SQL syntax and behavior.

If the option in the database is set to 2, it doesn’t matter what option you choose in this field.

- **Options**:
  - **Empty** - Gets the default from the database.
  - **1** - Old version of Firebird (1.5) > Does not support timestamp, Numeric or Decimal
  - **2** - Version with timestamp support
  - **3** - Version - Newer versions (Case sensitive and compatible with Firebird 2.5, 3 and 4)

Decimal Separator

Select the separator type for decimal records, between comma and period.

- **By default, period . is selected as the separator.**

Persistent connection

Set whether connections will be closed after your scripts run in Scriptcase applications.

- **By default, Scriptcase disables this option.**

Use schema before table name

Set whether the database schema is displayed before table names.

- **By default, Scriptcase enables this option.**
  - **Ex:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will

appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Firebird PDO

In Scriptcase, we have the following drivers available for Firebird connections: **Firebird PDO, Firebird.** If you are using your own pre-configured environment, the Firebird extensions must be enabled manually in PHP.

Prerequisites

If you are using a manual installation on MacOS, you will need to install PHP and the Firebird driver will be enabled. Click here to see how to do this.

1 - Check that the Firebird driver is enabled in your Scriptcase diagnostics. Here is how to locate your diagnostics below and verify that the driver is enabled.

- By going to the top menu **Help > Diagnostics**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a connection in Scriptcase

Here's how to create a connection in your Scriptcase project using the Firebird database.

1 - Access any project from your Scriptcase.

2 - Click on the **New connection** icon to create a connection.

or go to the **Database > New Connection** menu.

After that a new page will appear with all the database connections.

3 - Select the **Firebird** connection.

Connection

Enter the parameters for connecting to your Firebird database as follows:

- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix**conn** along with the database name.
- **DBMS Driver**: Select the Firebird Driver used to connect. In this example we use the**Firebird PDO** Driver.
- **Server/Host (Name or IP):** Enter the IP or domain of the server where the database is installed.
  - **Ex:** serverdomain.com or 192.168.254.170.
- **Port**: Enter the numeric port for your database server.
  - **Ex:** 3050
- **Database Name**: Enter the database created for use by the entered user.
- **Username**: Enter the user to authenticate the connection to your Firebird database.
- **Password:** Enter the password to complete the authentication process.
- **Test Connection**: Click this button to get a response from Scriptcase request to know if the entered parameters are correct.

Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.

Show

Allows the Firebird connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option will display your database views.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables are displayed. The setting can contain a PREFIX% or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

- **Owner:** Inform the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above.**
- **Show:** Choose whether tables are displayed for the owner entered.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to connection-specific settings. Changes made in this section impact the data display and performance of the application.

Charset

Select the encoding used in your database.

Role

Defines the user role for the database session. Roles are used to manage permissions and access control within Firebird.

You can set more than one role for the user separating them by using the “;”

Dialect

Determines the SQL dialect used in the connection. Firebird supports different dialects (1, 2, or 3), which affect SQL syntax and behavior.

If the option in the database is set to 2, it doesn’t matter what option you choose in this field.

- **Options**:
  - **Empty** - Gets the default from the database.
  - **1** - Old version of Firebird (1.5) > Does not support timestamp, Numeric or Decimal
  - **2** - Version with timestamp support
  - **3** - Version - Newer versions (Case sensitive and compatible with Firebird 2.5, 3 and 4)

Decimal Separator

Select the separator type for decimal records, between comma and period.

- **By default, period . is selected as the separator.**

Persistent connection

Set whether connections will be closed after your scripts run in Scriptcase applications.

- **By default, Scriptcase disables this option.**

Use schema before table name

Set whether the database schema is displayed before table names.

- **By default, Scriptcase enables this option.**
  - **Ex:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Informix in Scriptcase

In Scriptcase, we have the following driver available for connecting to Informix: **Informix PDO.** If you are using your own pre-configured environment, **the Informix extension must be enabled manually in PHP**.

You can verify that the driver is enabled by accessing your Scriptcase diagnostic. Here is how to locate your diagnostics.

- By accessing the top menu **Help > Diagnostics**, you can easily find it through the interface.

- Or, by accessing the url in your browser:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

If the driver is already enabled and you just want to know how to connect to Scriptcase, click on the link below according to the operating system that will be used.

Environment	Informix PDO
Windows	Conecte
Linux	Conecte
macOS	Conecte

Connection with Informix - Windows

In Scriptcase, we have the following driver available for connecting to Informix: **Informix PDO.** If you are using your own pre-configured environment, the Informix extension must be enabled manually in PHP.

NOTE: On automatic Scriptcase installation, the Informix PDO driver is already enabled. Just follow the steps to connect by clicking here.

Prerequisites

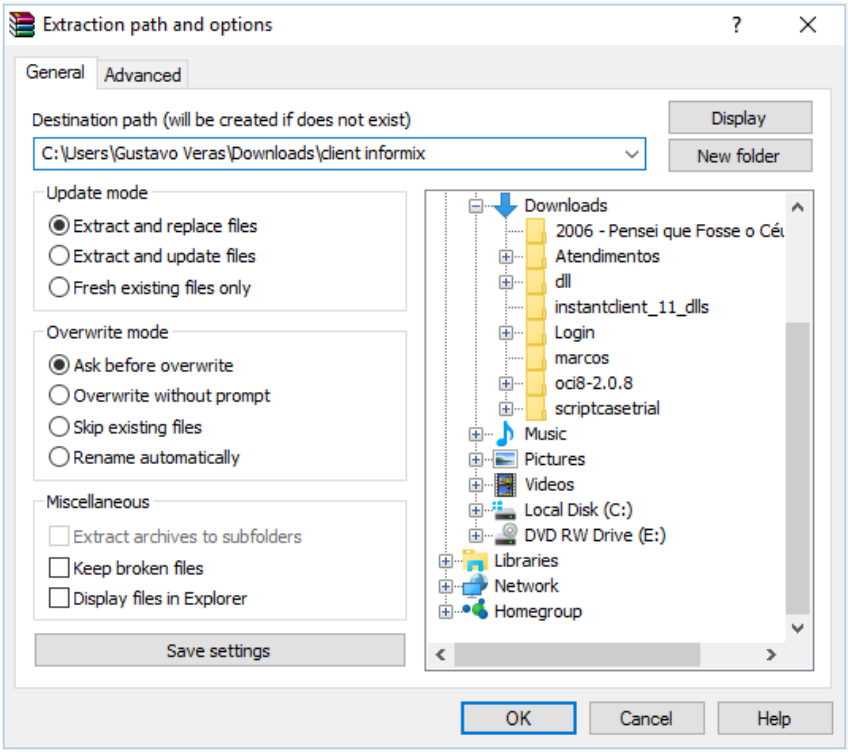
If you are using a manual installation on Windows, you will need to enable the Informix extension in the **php.ini** file and install the **Client SDK**. See below for how to do this.

- Download the instant client SDK compatible with your operating system's **x64** or **x86** architecture. Click here to access the Informix SDK Client Download page.
  - **x86 = 32 bits**
  - **x64 = 64 bits**

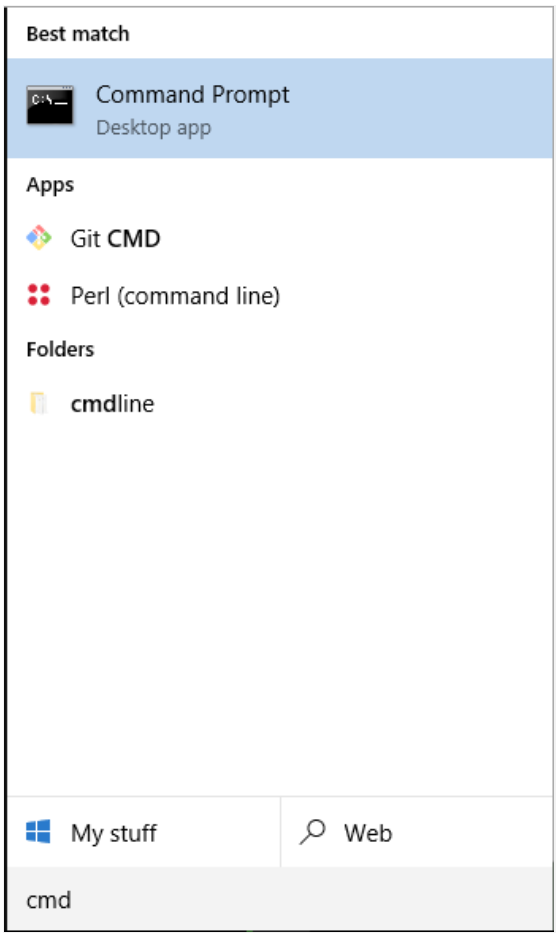
Informix SDK Client Installation

1 - Unzip the file into any folder.

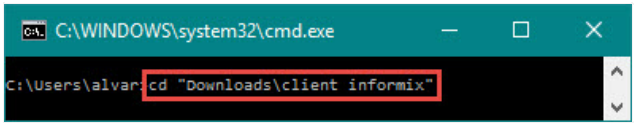
- **Example:** `client informix`



2 - Access your windows **CMD** by searching for command prompt in Windows search or by pressing **ctrl+r** and typing `cmd` in execute.

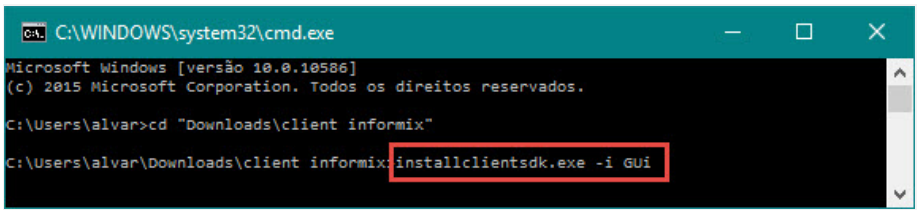


3 - When starting **CMD**, access the folder where the file was saved after downloading using the **cd** command and entering the path to the folder the SDK client is in. Example:

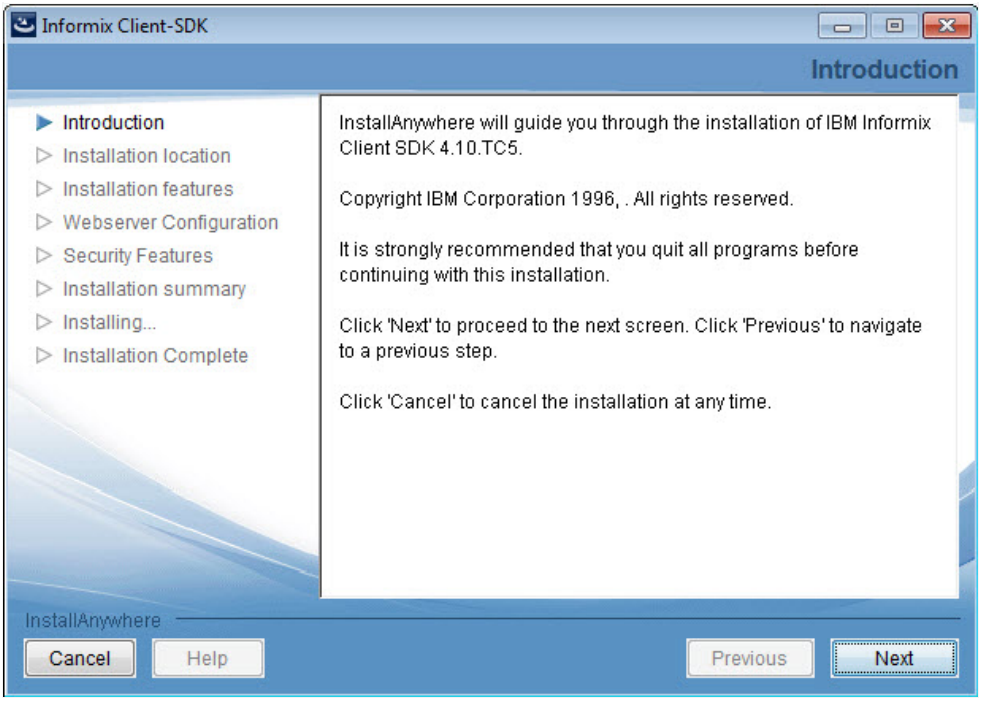


4 - Then run the .exe file with the `-i` and `GUI` parameters.

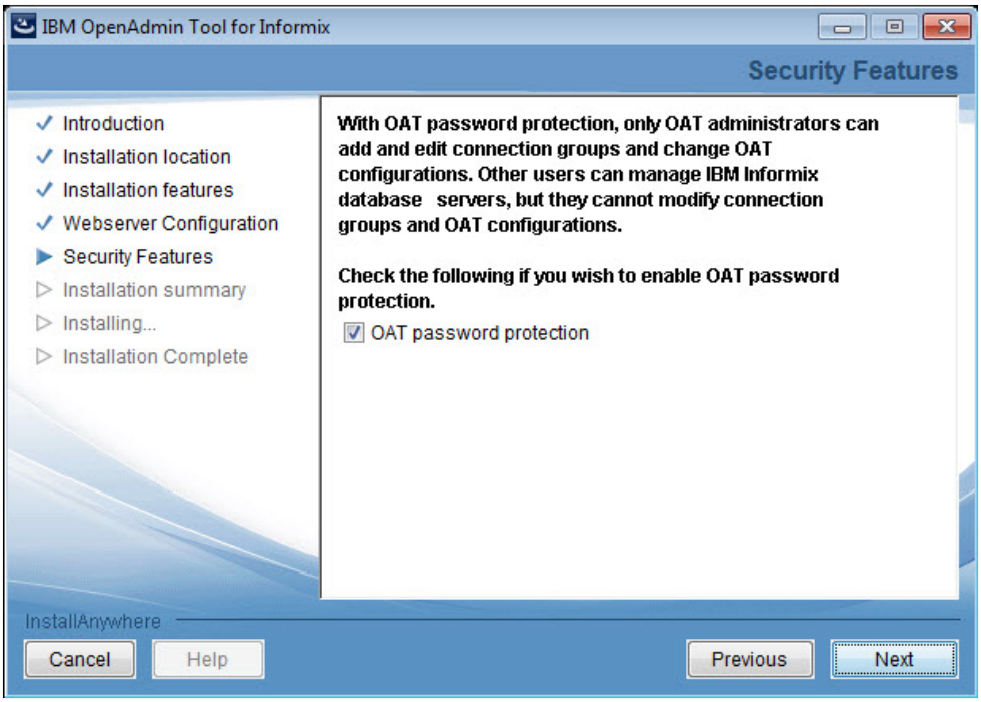
- **Example:** `installclientsdk.exe -i GUI`



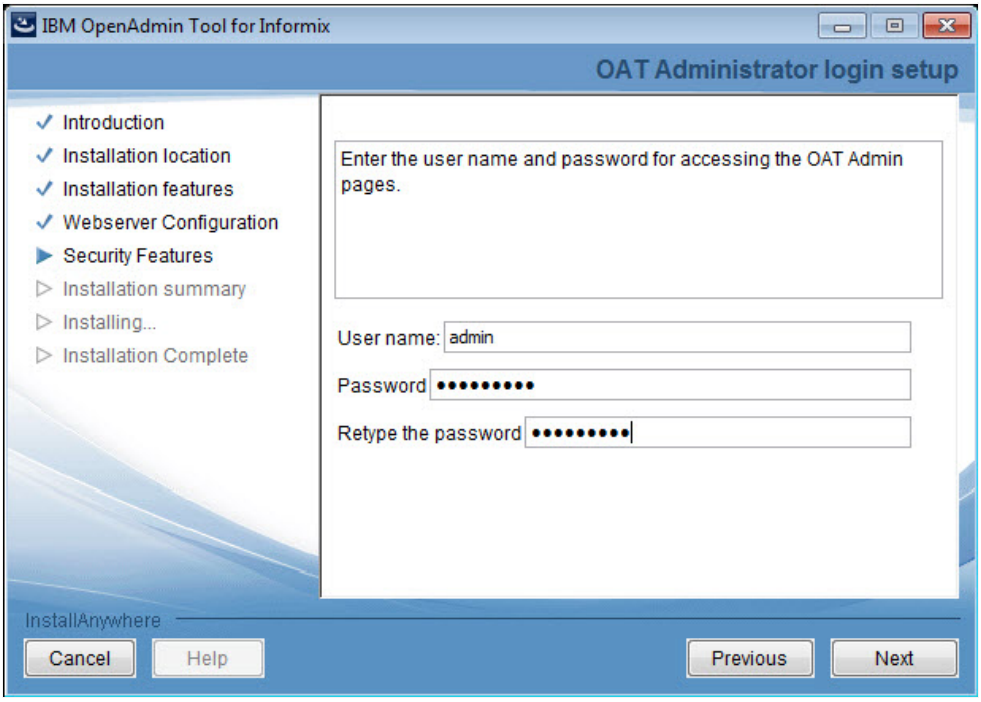
5 - Follow the client installation wizard by clicking next to keep the default values for the installation. Pay attention to the Security Features step.



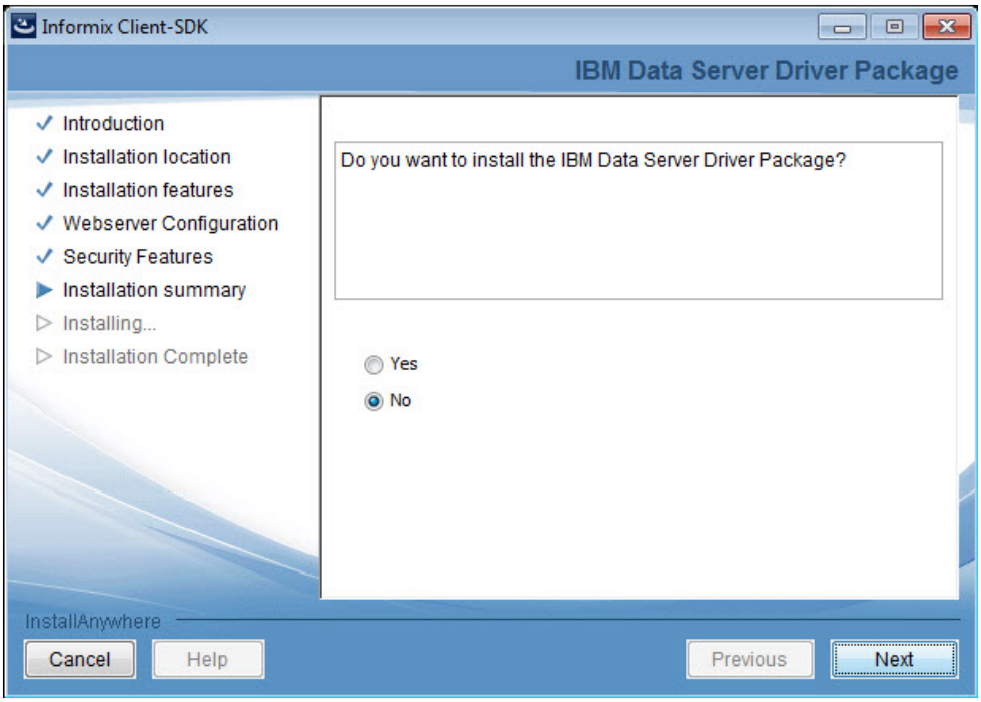
6 - Here you can choose whether or not to use a password to access the OAT (Informix Administration Application)



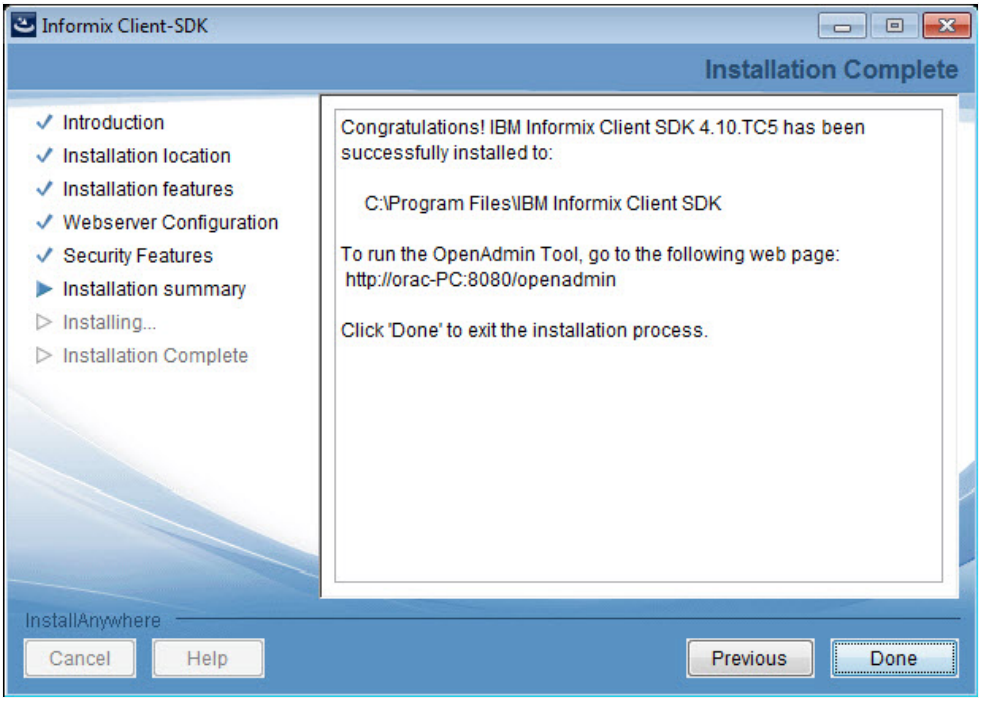
7 - Set your login and password, if you have checked the **OAT password protection** option



8 - Select **NO** and proceed with the installation as usual.



9 - Proceed with the installation by clicking on **Done** and wait for the process to finish.



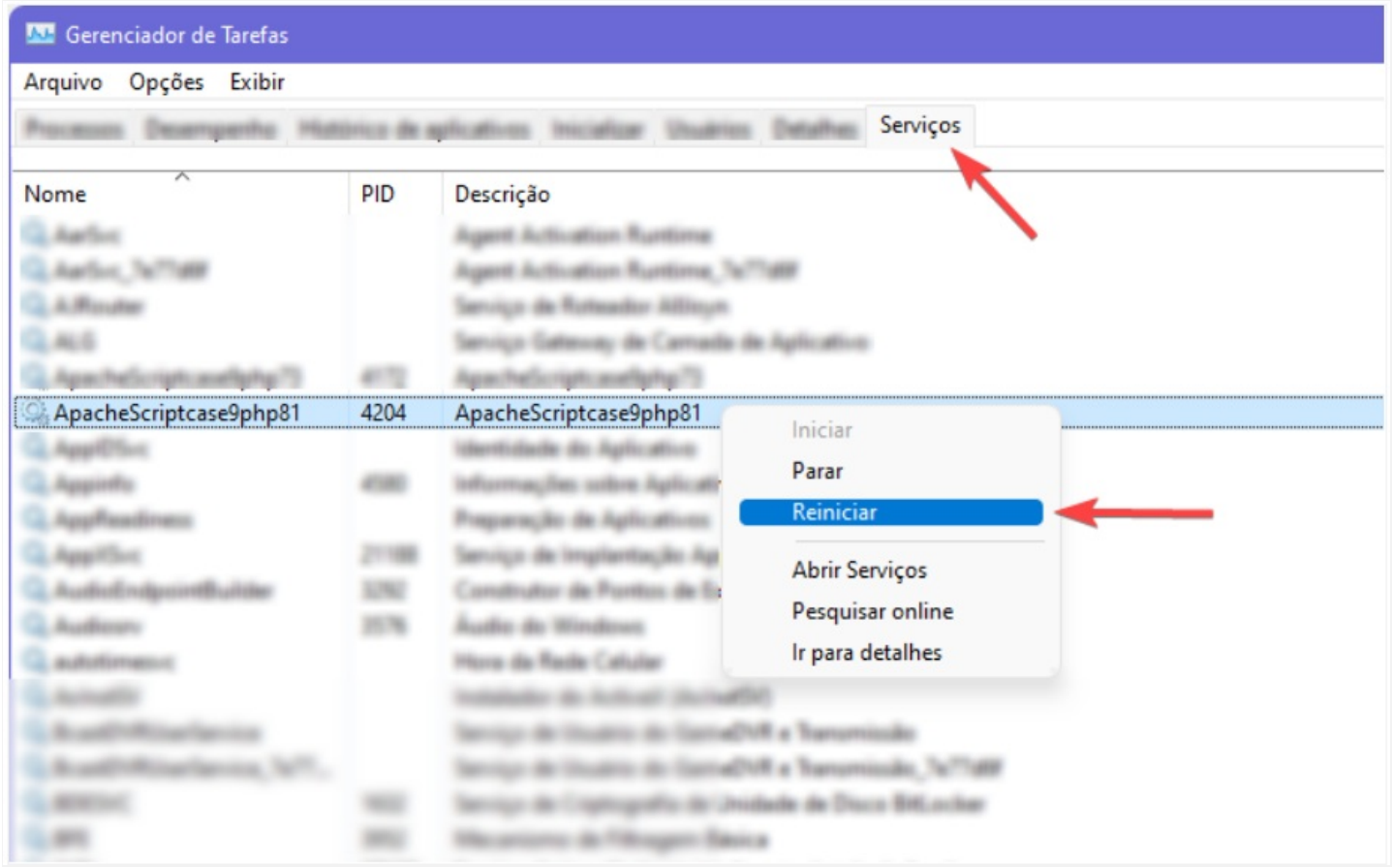
Setting `php.ini`.

1 - In the `php.ini` file, located under `C:\php`, uncomment the line for the Informix extension **php_pdo_informix** by removing the `;` from the start of the line. See the example below:

```
906  ;extension=bz2
907  ;extension=curl
908  ;extension=fileinfo
909  ;extension=gd2
910  ;extension=gettext
911  ;extension=gmp
912  ;extension=intl
913  ;extension=imap
914  extension=interbase
915  ;extension=ldap
916  ;extension=mbstring
917  ;extension=exif      ; Must be after mbstring as it depends on it
918  extension=mysqli
919  ;extension=oci8_12c  ; Use with Oracle Database 12c Instant Client
920  ;extension=odbc
921  ;extension=openssl
922  extension=pdo_firebird
923  extension=pdo_mysql
924  extension=pdo_oci
925  extension=pdo_odbc
926  extension=pdo_pgsql
927  extension=pdo_sqlite
928  extension=pgsql
929  ;extension=shmop
930  extension=php_pdo_informix
```

2 - Restart the Apache service using the **Task Manager**.

- Open Task Manager and click on the **Services** tab.
- Look for the `Apache2.4` or `ApacheScriptcase9php73` service and right-click on this service, then **Restart**.

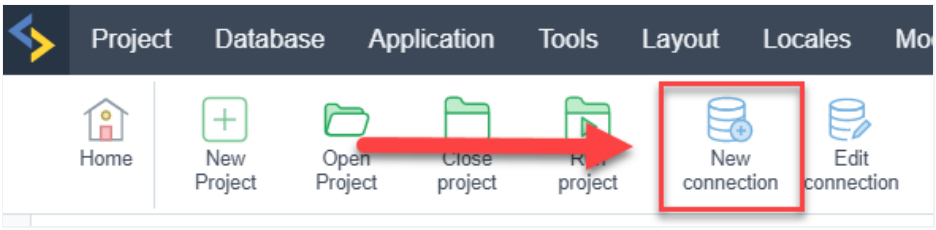


Creating a Connection in Scriptcase

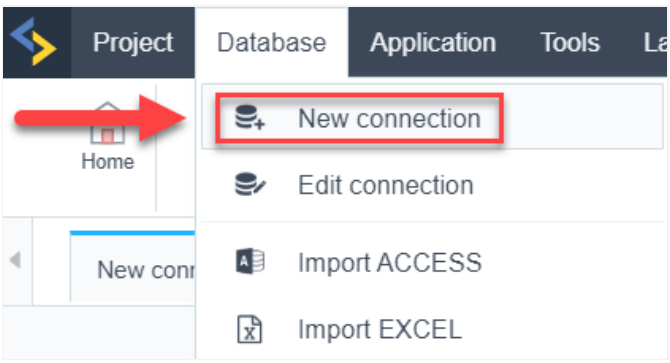
Here is how to create a connection in your Scriptcase project using the Informix database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

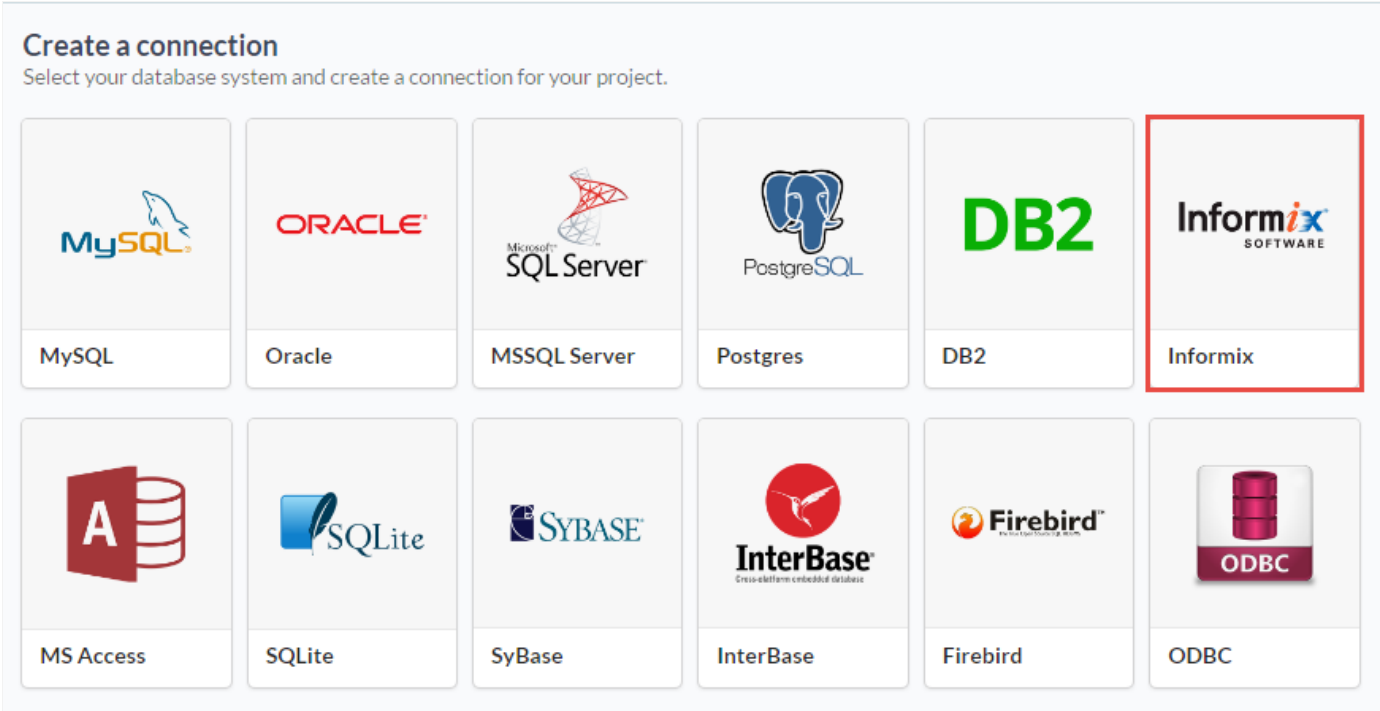


or go to the **Database > New Connection** menu.



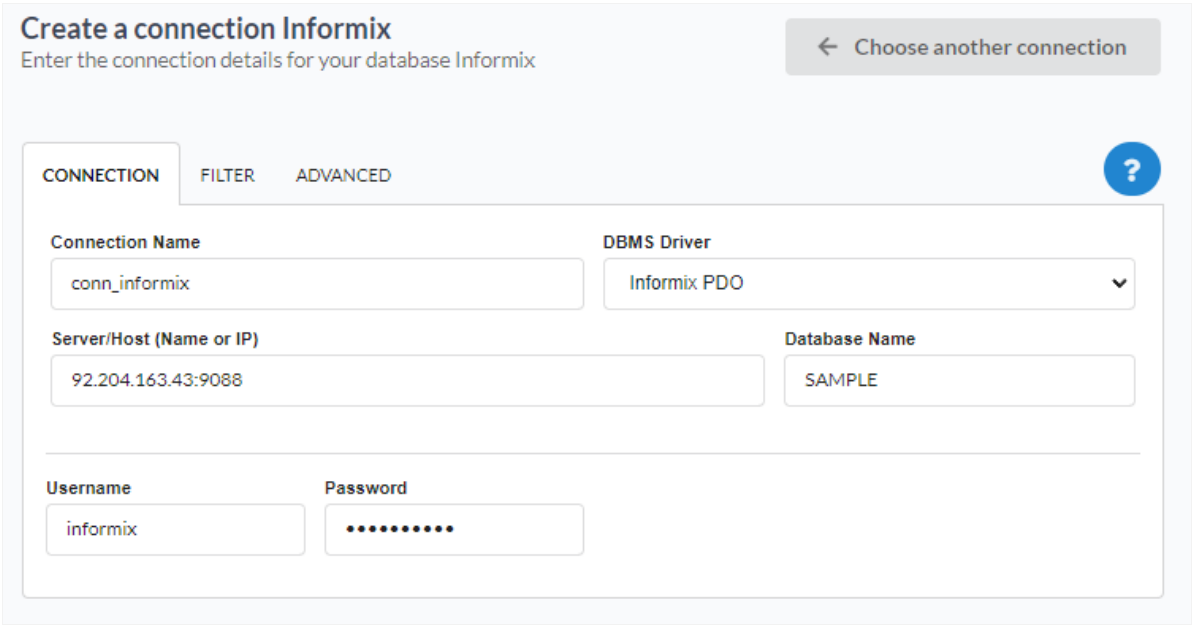
After that, a new page will appear with all the database connections.

3 - Select the connection **Informix**.



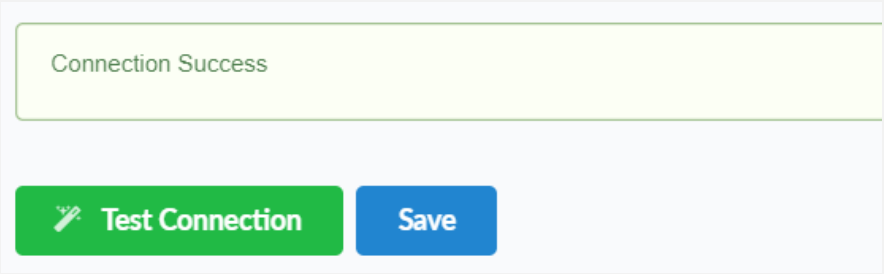
Connection

Enter the parameters for connecting to your Informix database as follows:



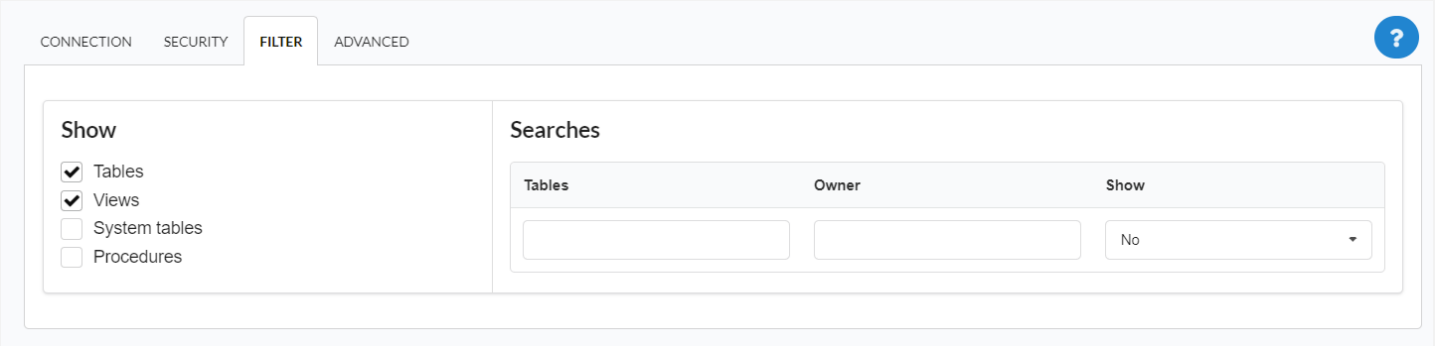
- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Informix Driver used to connect. In this example we use the **Informix PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with the access port.
  - **Ex:** `serverdomain.com:9088` or `192.168.254.170:9088`
- **Database Name**: Enter the name of your database to connect to it.
- **Username:** Enter the user to authenticate the connection to your Informix database.

- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click this button to get a response from Scriptcase request to know if the parameters entered are correct.



Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.



Show

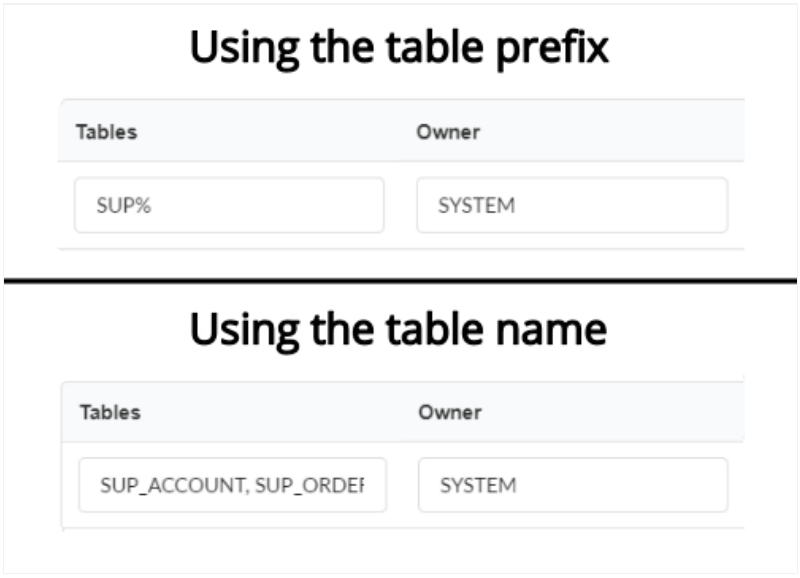
Allows the Informix connection to see tables, views, system tables, and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option will display your database views.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The setting can contain a `PREFIX%` or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

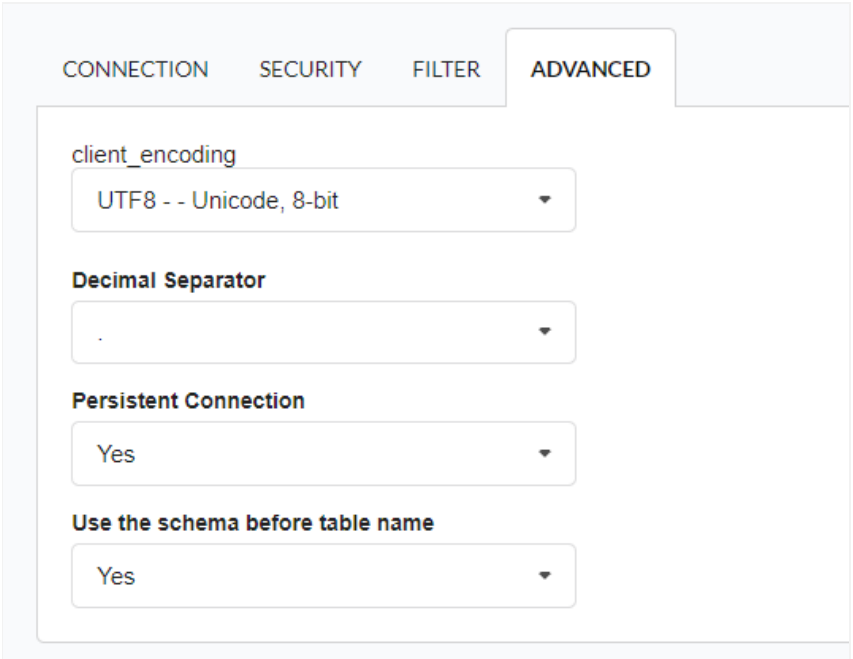


- **Owner:** Enter the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above.**
- **Show:** Choose whether tables are displayed for the owner entered.

NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to connection-specific settings. Changes made in this section impact the data display and performance of the application.

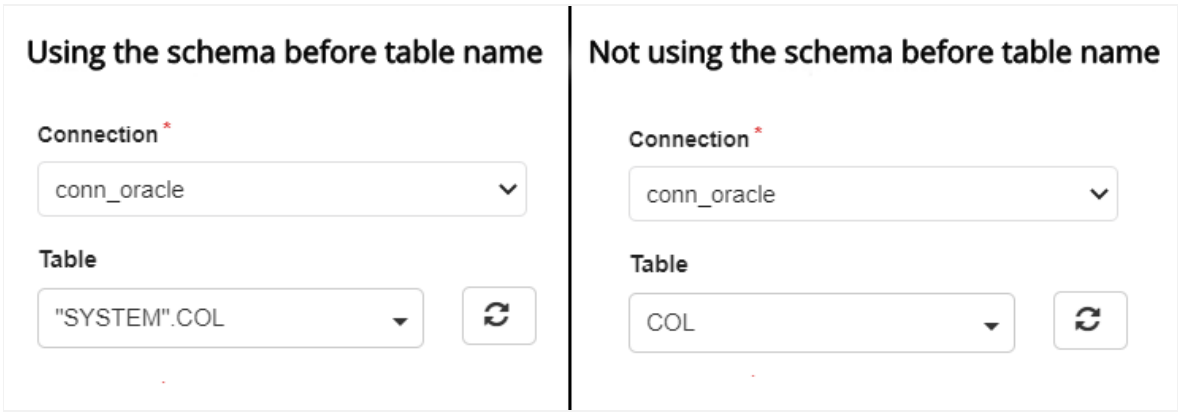


- **client_encoding:** Select the encoding used in your database. In the example above, we used client_encoding **UTF-8**.
- **Decimal separator:** Select the type of separator for decimal records, between comma and period.

- **By default, period `.` is selected as the separator.**

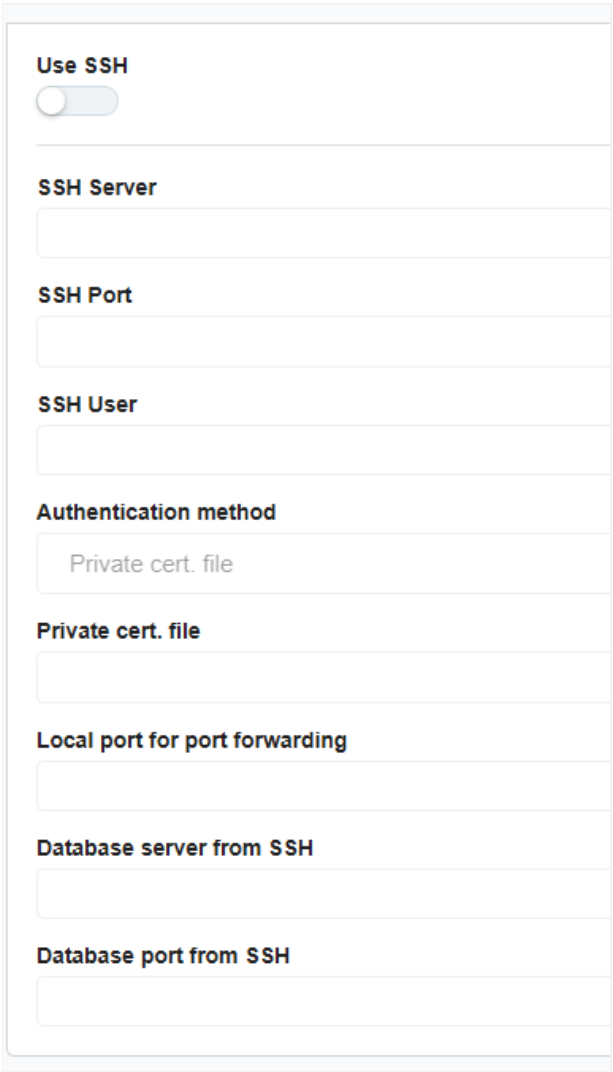
- **Persistent connection:** Set whether connections will be closed after your scripts run in Scriptcase applications.
  - **By default, Scriptcase disables this option.**

- **Use schema before table name:** Set whether the database schema is displayed before table names.
  - **By default, Scriptcase enables this option.**
  - **Ex:**



SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

Connection with Informix - Linux

In Scriptcase, we have the following driver available for connecting to Informix: **Informix PDO.** If you are using your own pre-configured environment, the Informix extension must be enabled manually in PHP.

NOTE: In the automatic installation of Scriptcase, the Informix PDO driver can be enabled if desired, during the installation process. If you have enabled it, just follow the steps to connect by clicking here. If you have not enabled it, see below how to enable the driver.

Enabling driver (Automatic installation)

1 - Download SDK Informix Driver by Clicking here

2 - Go to the directory **/opt/Scriptcase/v9-php81/components/drivers/** and create the following folders: **IBM/informix**

The full path should be: **/opt/Scriptcase/v9-php81/components/drivers/IBM/informix**

3 - Extract the files inside the folder **informix**

4 - Enter the **informix** folder through the terminal and run the command below

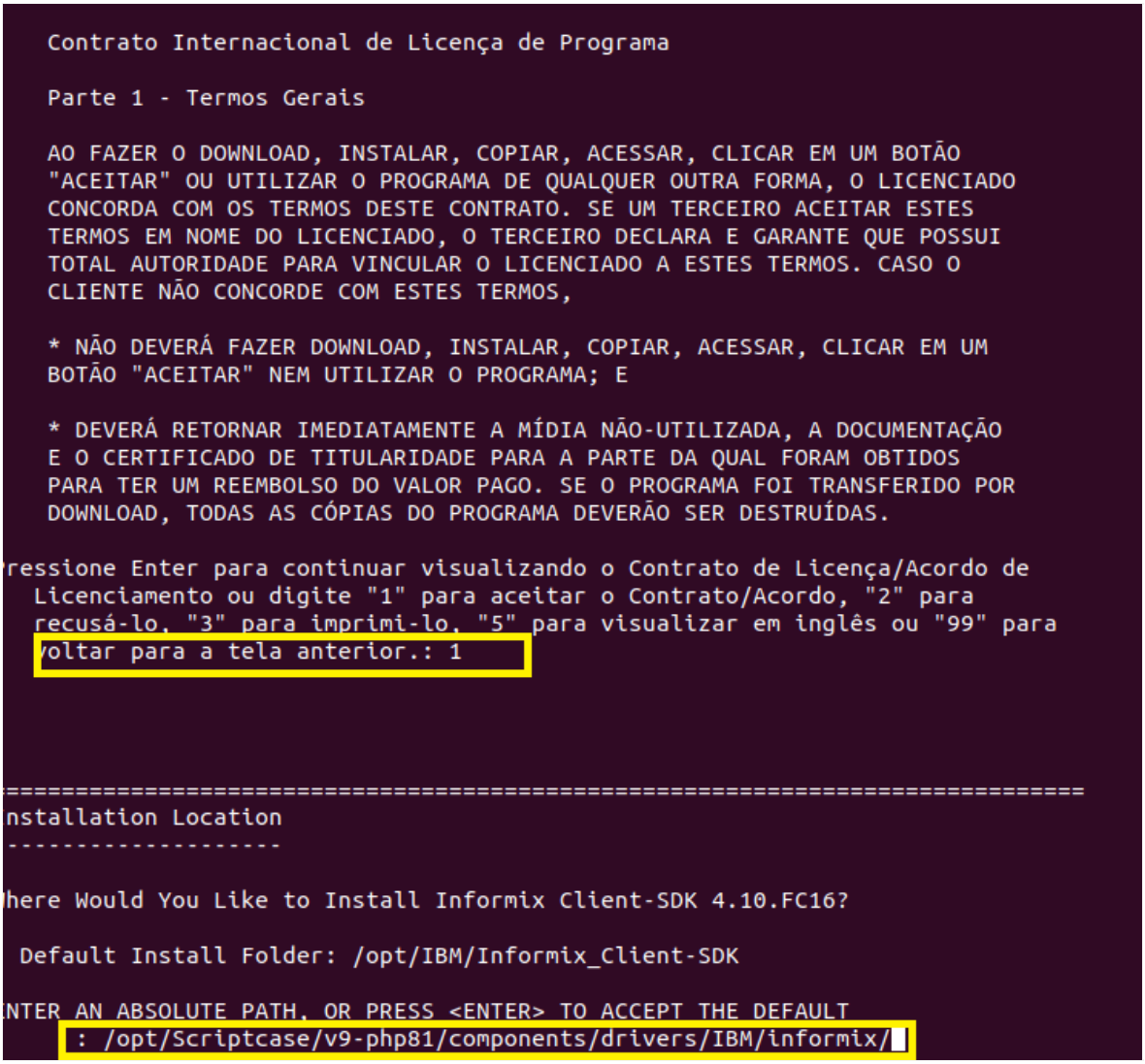
```
sudo tar -xf clientsdk.4.10.FC16_672.LINUX.tar
```

5 - Run the installation file using the command below:

```
sudo ./installclientsdk
```

6 - When entering the installation process, type **1** to accept the terms and then type the absolute path of the driver folder and press ENTER.

The absolute path must be: **/opt/Scriptcase/v9-php81/components/drivers/IBM/informix** as shown in the image below



7 - Then choose the features that will be installed by the installer. Choose all features as shown in the image and press ENTER to proceed

```
================================================================================
Choose ClientSDK Features to Install
---------------------------------

ENTER A COMMA_SEPARATED LIST OF NUMBERS REPRESENTING THE FEATURES YOU WOULD
LIKE TO SELECT, OR DESELECT. TO VIEW A FEATURE'S DESCRIPTION, ENTER
'?<NUMBER>'.  PRESS <RETURN> WHEN YOU ARE DONE:

    1- [X] Informix Client-SDK
    2-  |-[X] Informix Object Interface for C++
    3-     |-[X] Informix Object Interface for C++ Demos
    4-  |-[X] Informix ESQL/C
    5-     |-[X] Informix ESQL/C Demos
    6-     |-[X] 7.2 Application Compatibility Module
    7-  |-[X] Informix LIBDMI for Client Applications
    8-  |-[X] Informix ODBC Driver
    9-     |-[X] Informix ODBC Driver Demos
   10-  |-[X] Informix Common Database Utilities
   11- [X] Global Language Support (GLS)
   12-  |-[X] West European and Americas
   13-  |-[X] East European and Cyrillic
   14-  |-[X] Chinese
   15-  |-[X] Japanese
   16-  |-[X] Korean
   17-  |-[X] Thai

Press <ENTER> to install above selected features or choose the corresponding
number to change the feature selection: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17
```

8 - Wait for the installation to complete. You should see information that the installation was completed successfully.

```
================================================================================
Installation Complete
---------------------

Congratulations. Informix Client-SDK 4.10.FC16 has been successfully installed
to:

/opt/Scriptcase/v9-php81/components/drivers/IBM/informix


PRESS <ENTER> TO EXIT THE INSTALLER:
root@jader-VirtualBox:/opt/Scriptcase/v9-php81/components/drivers/IBM/informix#
```

9 - Restart the Apache

sudo service apachesc9php81 restart

Enabling driver (Manual installation)

Prerequisites

Install the dependencies below:

UBUNTU\DEBIAN	CENTOS\RHEL
- sudo apt-get update	- sudo yum update
- sudo apt-get install libaio1	- sudo yum install libaio-devel
- sudo apt-get install libncurses5	- sudo yum install libaio
- sudo apt-get install alien	- sudo yum install glibc
- sudo apt-get install gcc-multilib g++-multilib	- sudo yum install compat-libstdc++-33
- sudo apt-get install libpam0g	- sudo yum install glibc-devel
- sudo apt-get install ksh	- sudo yum install libstdc++
- sudo apt-get install unixodbc-dev unixodbc	- sudo yum install libstdc++
	- sudo yum install pam
	- sudo yum install ncurses-devel
	- sudo yum install unixODBC
	- sudo wget ftp://195.220.108.108/linux/centos/6.7/os/i386/Packages/ksh-20120801-28.el6.i686.rpm
	- sudo yum install ksh-20120801-28.el6.i686.rpm
	- sudo yum install nano

Informix SDK Client Configuration and Installation

1 - Download the instant client SDK compatible with your operating system's **x64** or **x86** architecture. Click here to access the Informix SDK Client Download page.

- **x86 = 32 bits**
- **x64 = 64 bits**

NOTE: The Informix client installation directory must be according to your architecture:

- **32 Bits:** /opt/IBM/ifx
- **64 Bits:** /opt/IBM/x64/informix

2 - Realize a instalação do Client SDK:

x86 Architecture	x64 Architecture
1 - sudo tar -xf x86_clientsdk.3.70.UC8DE.LINUX.tar	1 - sudo tar -xf x64_clientsdk.4.10.FC6DE.LINUX.tar
2 - sudo ./installclientsdk	2 - sudo ./installclientsdk
3 - Select 1 to accept the terms.	3 - Select 1 to accept the terms.
4 - Select the options: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17	4 - Select the options: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17

Proceed with completing the installation.

3 - Restart the Apache service using the following command in the terminal:

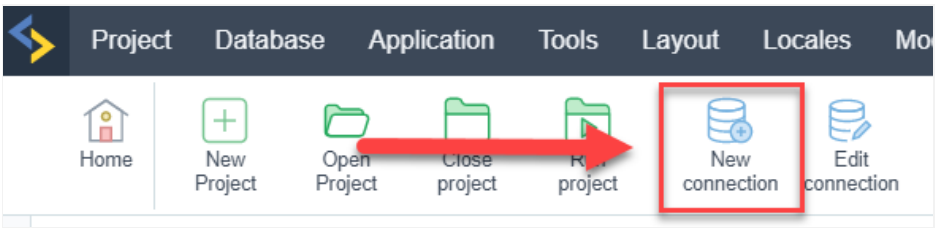
Ubuntu 20\Debian 9 or Higher	CentOS 7/8 or Higher
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

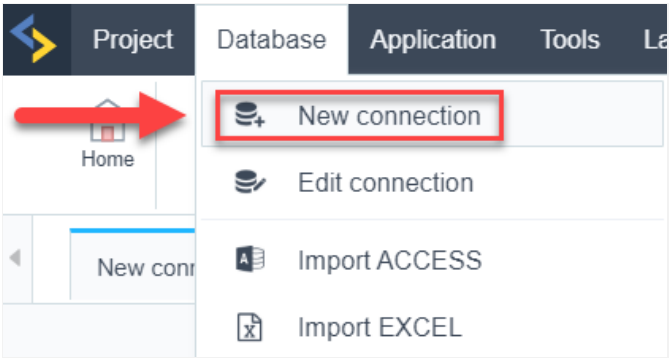
Here is how to create a connection in your Scriptcase project using the Informix database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

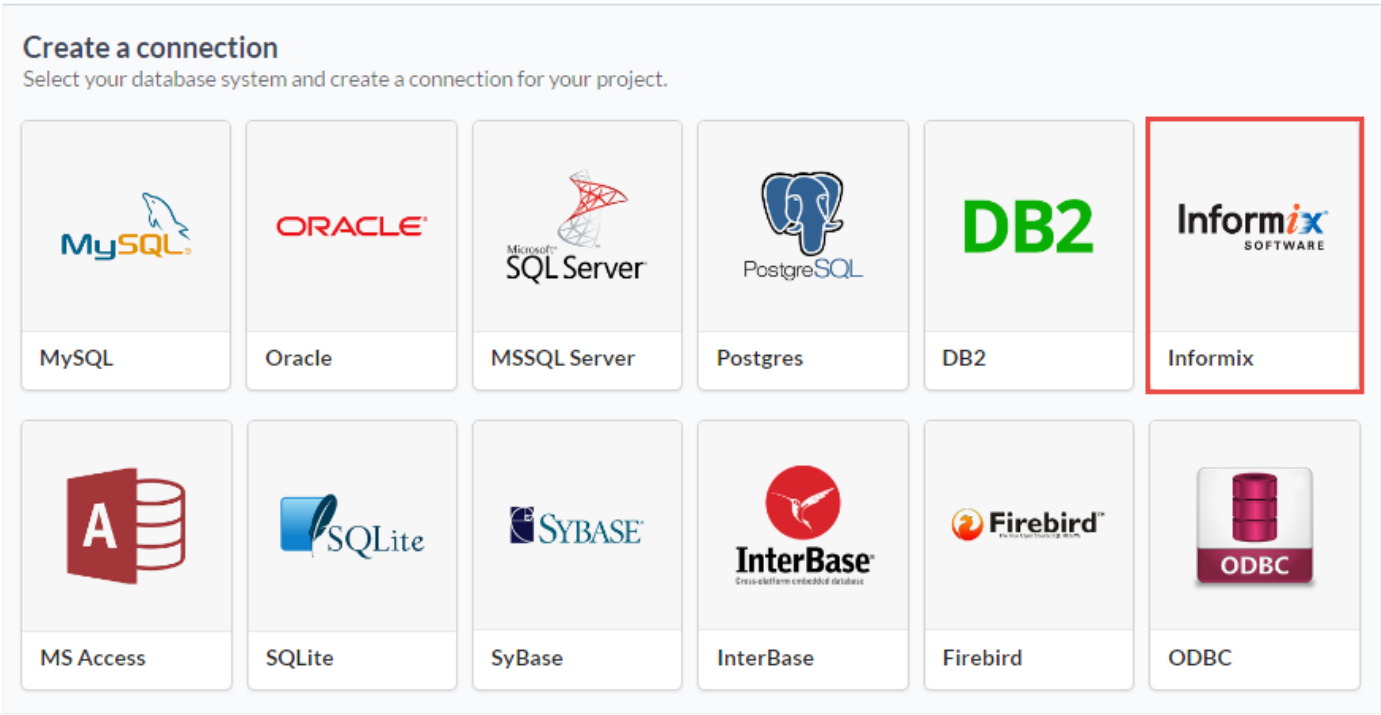


or go to the **Database > New Connection** menu.



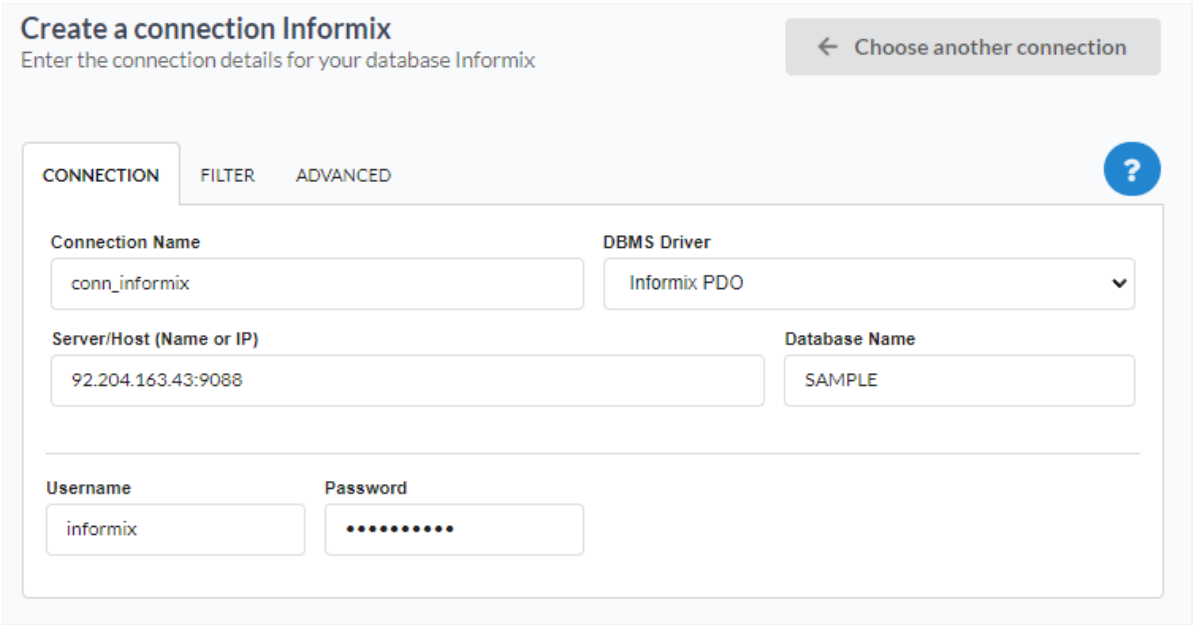
After that, a new page will appear with all the database connections.

3 - Select the connection **Informix**.



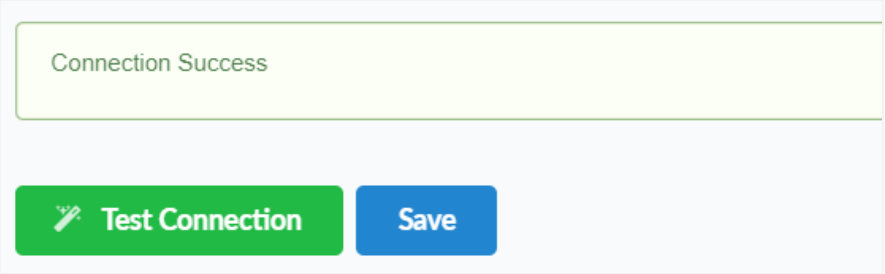
Connection

Enter the parameters for connecting to your Informix database as follows:



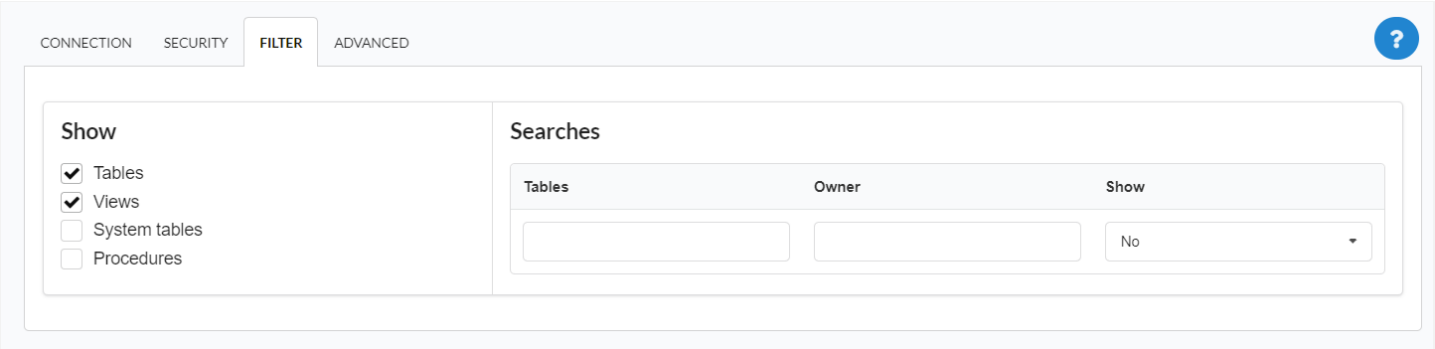
- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Informix Driver used to connect. In this example we use the **Informix PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with the access port.
  - **Ex:** `serverdomain.com:9088` or `192.168.254.170:9088`
- **Database Name**: Enter the name of your database to connect to it.
- **Username:** Enter the user to authenticate the connection to your Informix database.
- **Password**: Enter the password to complete the authentication process.

- **Test Connection**: Click this button to get a response from Scriptcase request to know if the parameters entered are correct.



Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.



Show

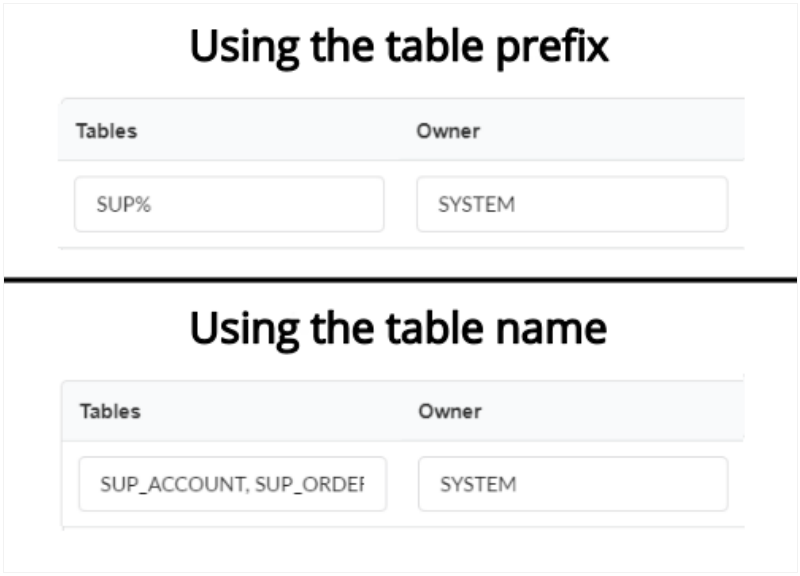
Allows the Informix connection to see tables, views, system tables, and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option will display your database views.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The setting can contain a `PREFIX%` or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

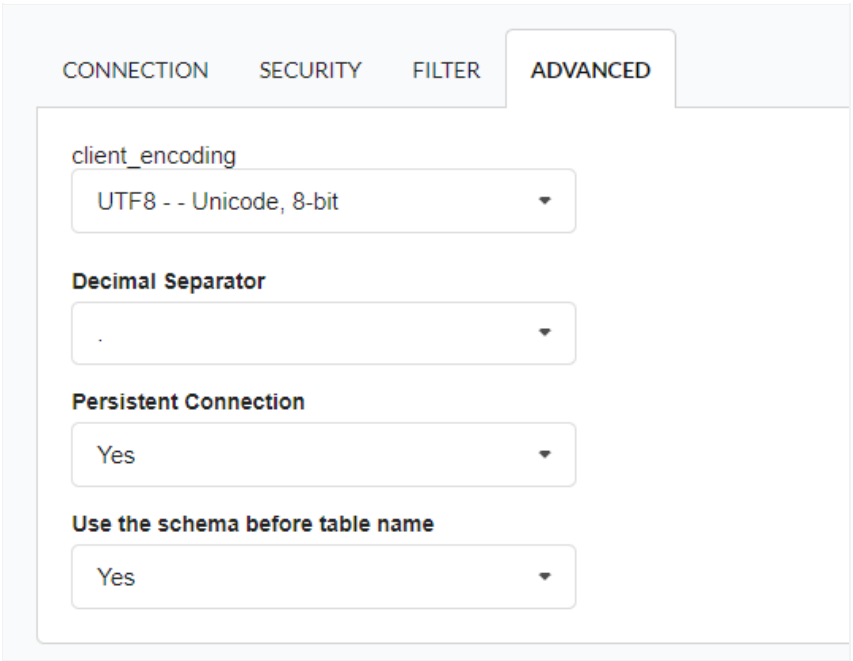


- **Owner:** Enter the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above.**
- **Show:** Choose whether tables are displayed for the owner entered.

NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

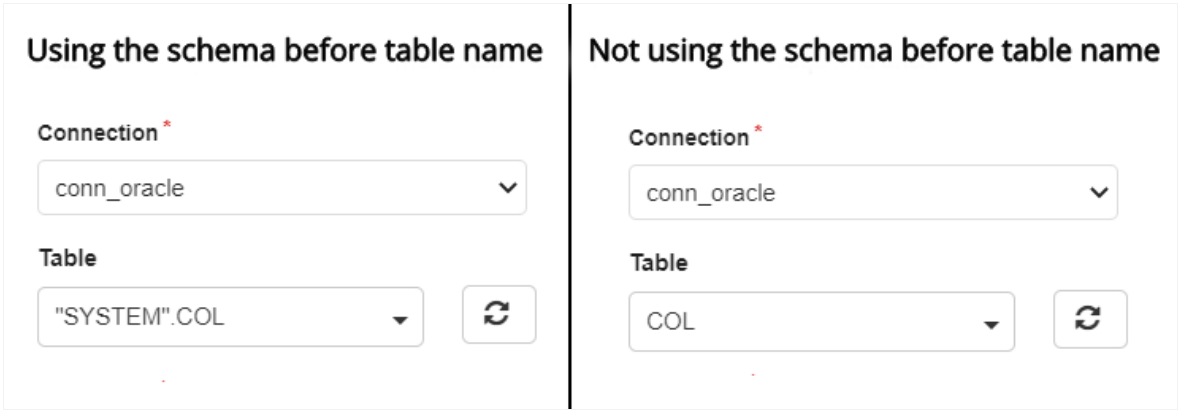
In this tab, you have access to connection-specific settings. Changes made in this section impact the data display and performance of the application.



- **client_encoding:** Select the encoding used in your database. In the example above, we used client_encoding **UTF-8**.
- **Decimal separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, period `.` is selected as the separator.**

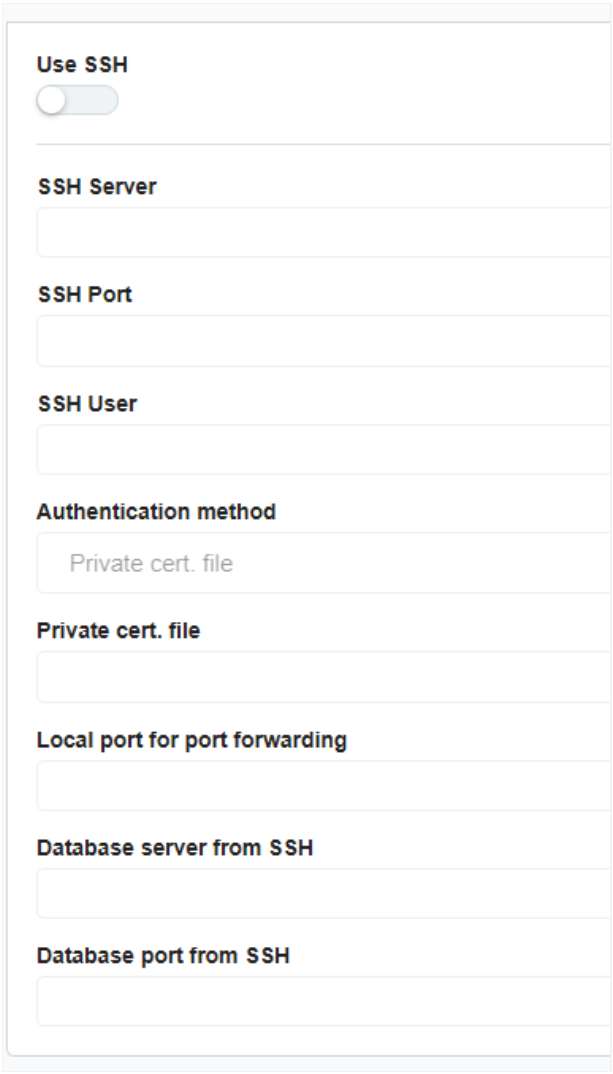
- **Persistent connection:** Set whether connections will be closed after your scripts run in Scriptcase applications.
  - **By default, Scriptcase disables this option.**

- **Use schema before table name:** Set whether the database schema is displayed before table names.
  - **By default, Scriptcase enables this option.**
  - **Ex:**



SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

Connection with Informix - MacOs

In Scriptcase, we have the following driver available for connecting to Informix: **Informix PDO.** If you are using your own pre-configured environment, the Informix extension must be enabled manually in PHP.

NOTE: On automatic Scriptcase installation, the Informix PDO driver is already enabled. Just follow the steps to connect by clicking here.

Prerequisites

- Download the instant client SDK compatible with your operating system's **x64** or **x86** architecture. Click here to access the Informix SDK Client Download page.
  - **x86 = 32 bits**
  - **x64 = 64 bits**

Informix SDK Client Configuration and Installation

1 - Extract the instant Client SDK file.

2 - Perform the installation via terminal, accessing the extracted folder and running the command below:

```
sudo ./installclientsdk
```

NOTE: Install in the default directory (/Applications/IBM/informix).

3 - Choose options 1 through 17 and finish the installation.

4 - After doing so, save the file and restart the apache service.

MacOs "Catalina" 10.15 or Higher

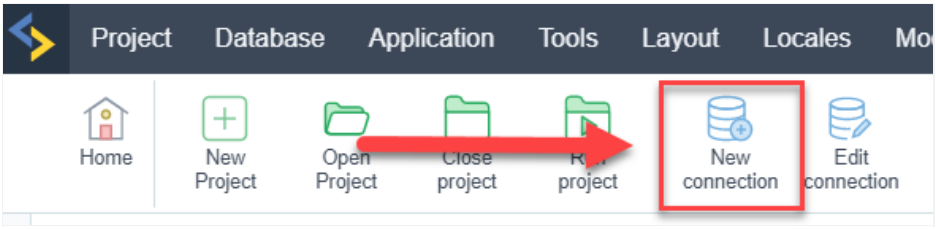
```
sudo apachectl restart
```

Creating a Connection in Scriptcase

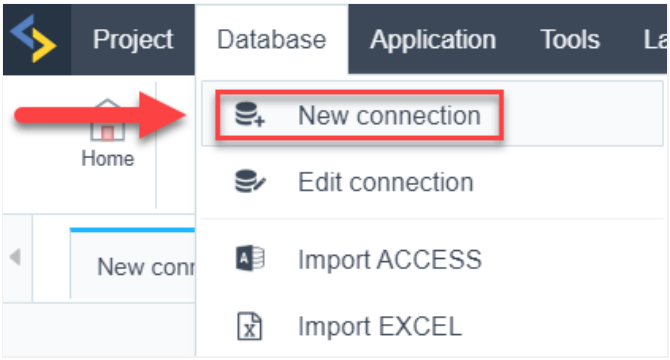
Here is how to create a connection in your Scriptcase project using the Informix database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

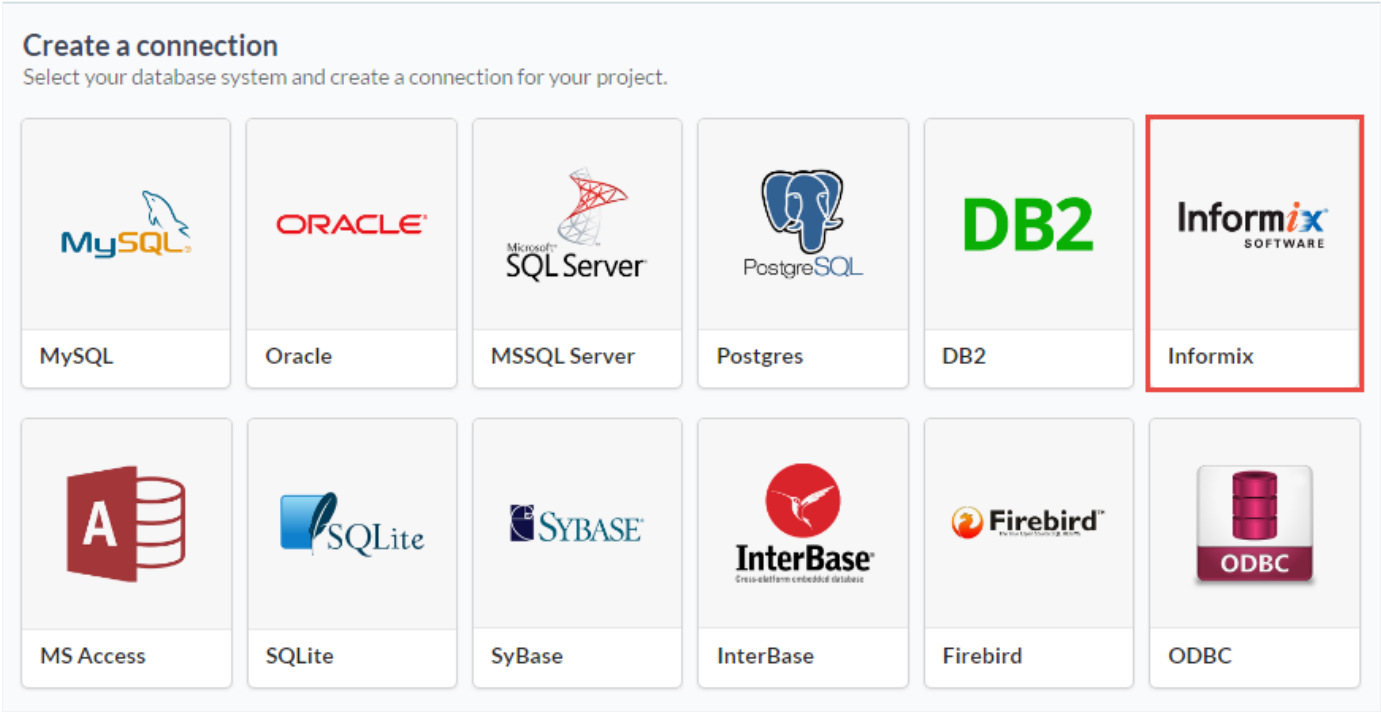


or go to the **Database > New Connection** menu.



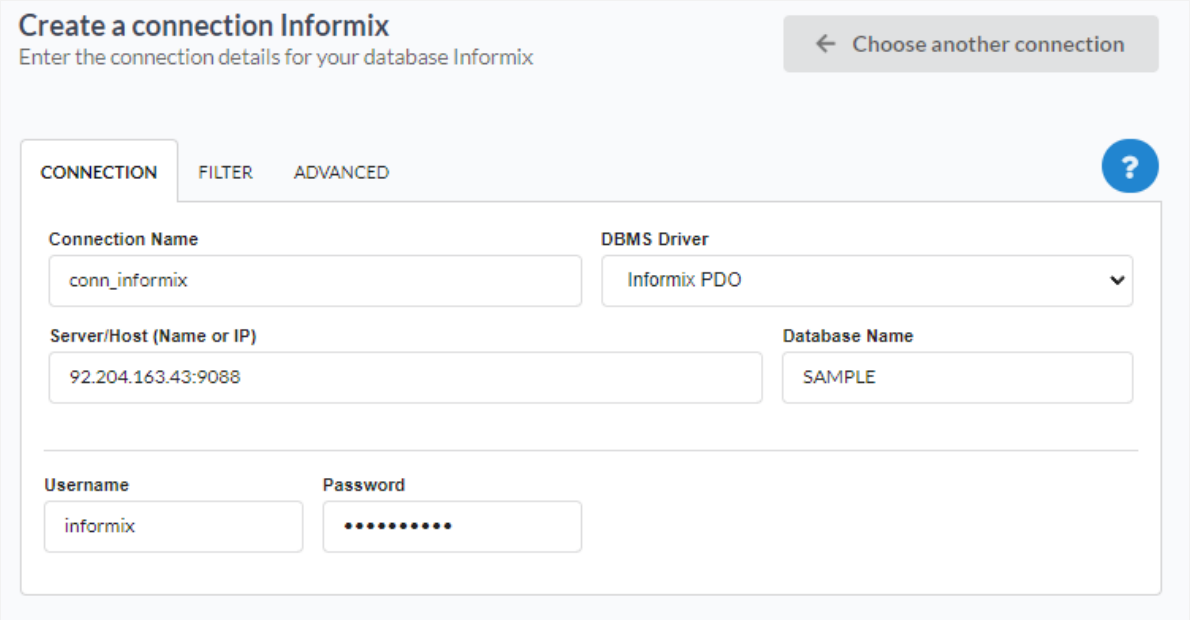
After that, a new page will appear with all the database connections.

3 - Select the connection **Informix**.

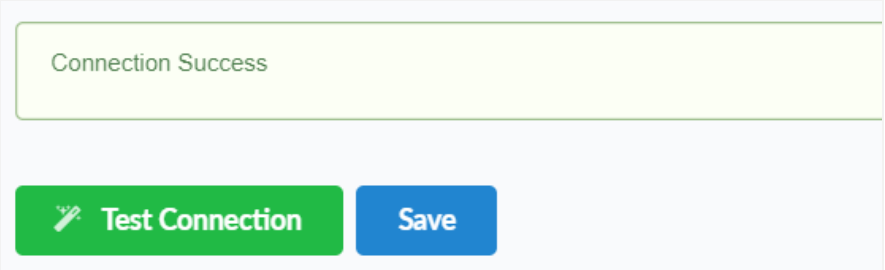


Connection

Enter the parameters for connecting to your Informix database as follows:

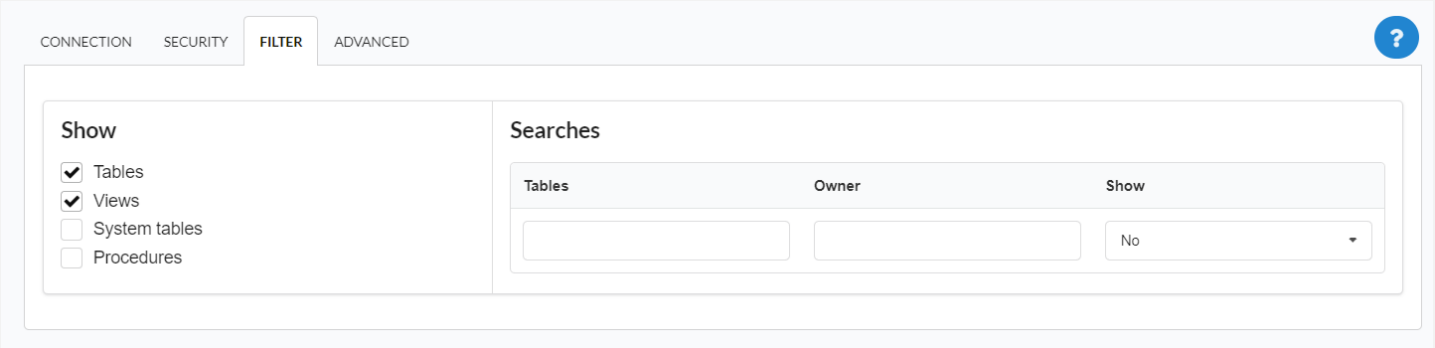


- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Informix Driver used to connect. In this example we use the **Informix PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with the access port.
  - **Ex:** `serverdomain.com:9088` or `192.168.254.170:9088`
- **Database Name**: Enter the name of your database to connect to it.
- **Username:** Enter the user to authenticate the connection to your Informix database.
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click this button to get a response from Scriptcase request to know if the parameters entered are correct.



Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.



Show

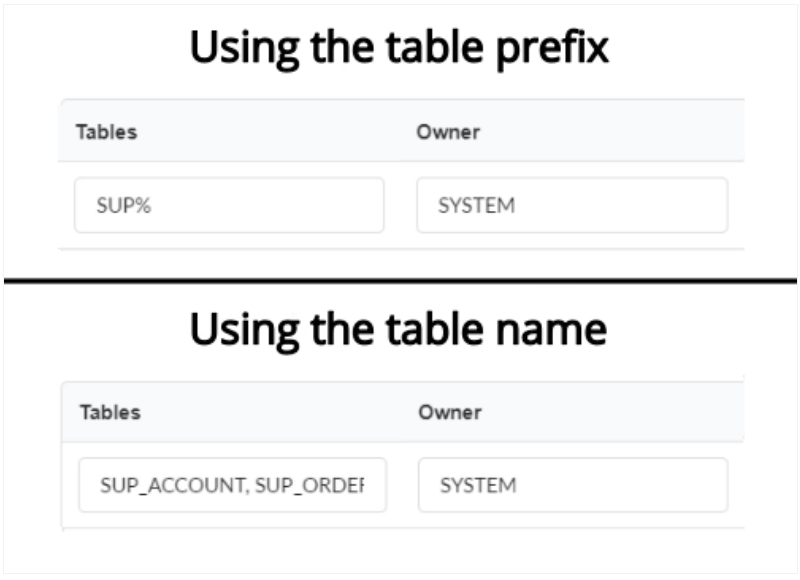
Allows the Informix connection to see tables, views, system tables, and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option will display your database views.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The setting can contain a `PREFIX%` or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**



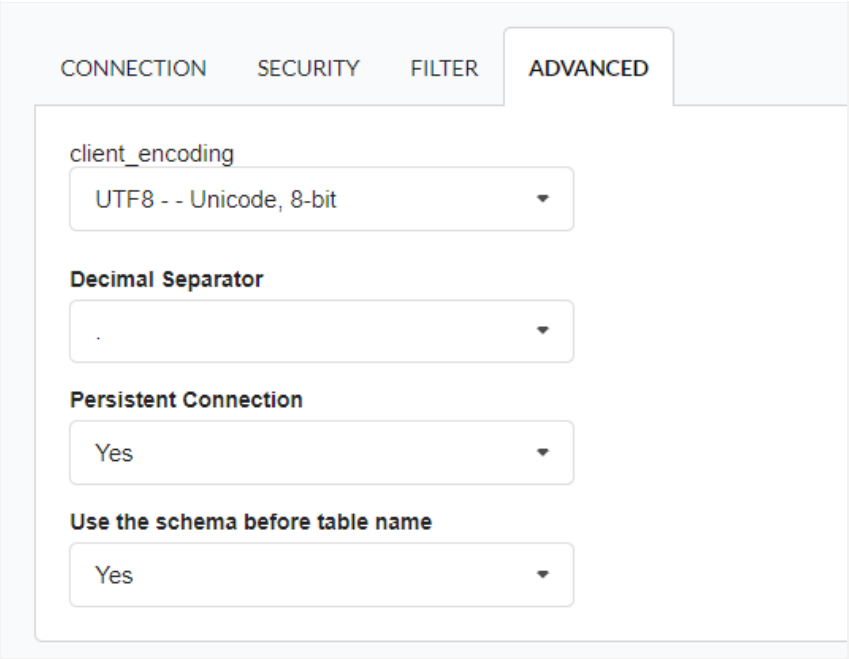
- **Owner:** Enter the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above.**

- **Show:** Choose whether tables are displayed for the owner entered.

NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to connection-specific settings. Changes made in this section impact the data display and performance of the application.

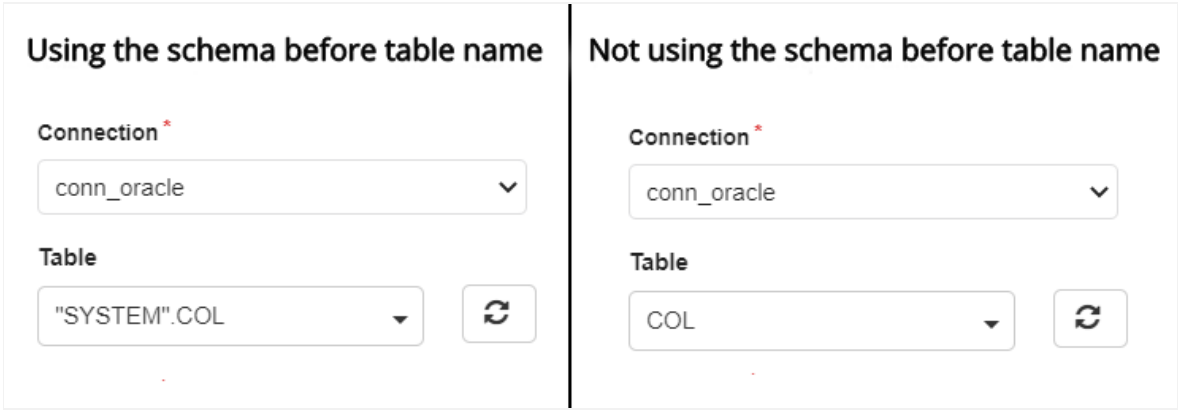


- **client_encoding:** Select the encoding used in your database. In the example above, we used client_encoding **UTF-8**.
- **Decimal separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, period `.` is selected as the separator.**

- **Persistent connection:** Set whether connections will be closed after your scripts run in Scriptcase applications.
  - **By default, Scriptcase disables this option.**

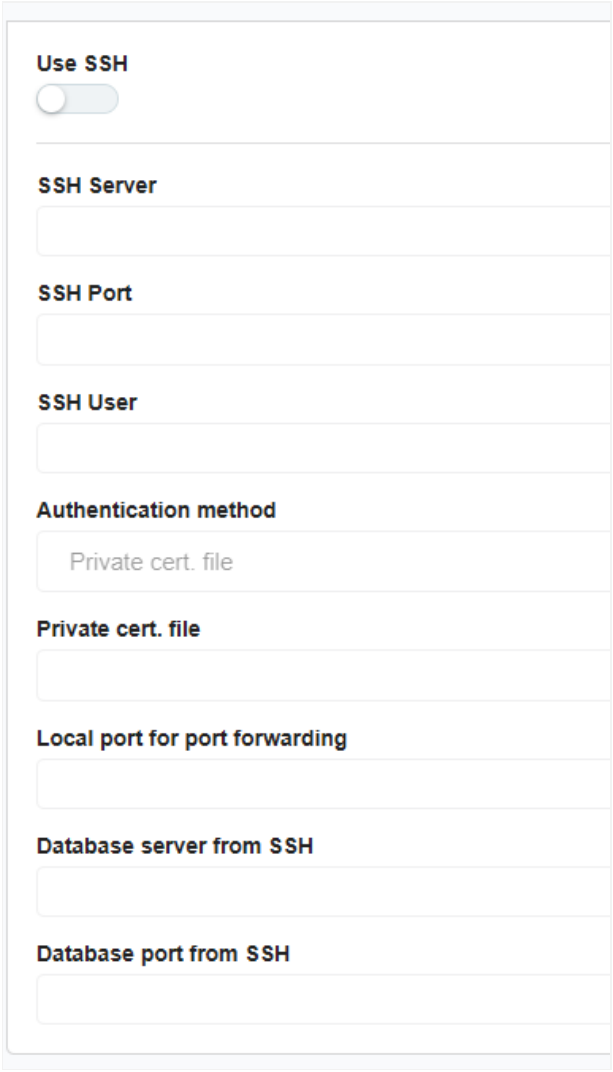
- **Use schema before table name:** Set whether the database schema is displayed before table names.
  - **By default, Scriptcase enables this option.**

  - **Ex:**



SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

Interbase with Scriptcase

In Scriptcase, we have only one driver to connect with Interbase: **pdo_firebird**. If you are using your own pre-configured environment, PDO Firebird extension must be manually enabled in PHP.

You can check your enabled drivers by accessing your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

Connections available

Firebird connection is available for Windows, Linux and MacOS with Firebird PDO driver.

Select the system where Scriptcase will be used and see our documentation about the connection.

- Interbase on Windows
- Interbase on Linux
- Interbase on macOS

Interbase connection with PDO Firebird

In Scriptcase, we have the following driver available for Interbase connections: **PDO Firebird.** If you are using your own pre-configured environment, Interbase extensions must be enabled manually in PHP.

Prerequisites

If you are using a manual installation on Windows, you will need to enable the PDO Firebird extension in the **php.ini** file. See below for how to do this.

1 - In the php.ini file, located under C:\php, uncomment the line for Firebird extension: **pdo_firebird** by removing the ; from the beginning of the line. See the example below:

2 - Restart the Apache service using the **Task Manager**.

- Open Task Manager and click on the **Services** tab.
- Look for the Apache2.4 or ApacheScriptcase9php81 service and right-click on this service, then **Restart**.

Creating a connection in Scriptcase

Here's how to create a connection in your Scriptcase project using the Firebird database.

1 - Access any project from your Scriptcase.

2 - Click on the **New connection** icon to create a connection.

or go to the **Database > New Connection** menu.

After that a new page will appear with all the database connections.

3 - Select the **Firebird** connection.

Connection

Enter the parameters for connecting to your Firebird database as follows:

- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Firebird Driver used to connect. In this example we use the **Firebird PDO** Driver.
- **Server/Host (Name or IP):** Enter the IP or domain of the server where the database is installed.
  - **Ex:** serverdomain.com or 192.168.254.170.
- **Port**: Enter the numeric port for your database server.
  - **Ex:** 3050
- **Database Name**: Enter the database created for use by the entered user.
- **Username**: Enter the user to authenticate the connection to your Firebird database.
- **Password:** Enter the password to complete the authentication process.
- **Test Connection**: Click this button to get a response from Scriptcase request to know if the entered parameters are correct.

Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.

Show

Allows the Firebird connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option will display your database views.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables are displayed. The setting can contain a PREFIX% or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

- **Owner:** Inform the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above.**
- **Show:** Choose whether tables are displayed for the owner entered.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to connection-specific settings. Changes made in this section impact the data display and performance of the application.

Charset

Select the encoding used in your database.

Role

Defines the user role for the database session. Roles are used to manage permissions and access control within Firebird.

You can set more than one role for the user separating them by using the “;”

Dialect

Determines the SQL dialect used in the connection. Firebird supports different dialects (1, 2, or 3), which affect SQL syntax and behavior.

If the option in the database is set to 2, it doesn’t matter what option you choose in this field.

- **Options**:
  - **Empty** - Gets the default from the database.
  - **1** - Old version of Firebird (1.5) > Does not support timestamp, Numeric or Decimal
  - **2** - Version with timestamp support
  - **3** - Version - Newer versions (Case sensitive and compatible with Firebird 2.5, 3 and 4)

Decimal Separator

Select the separator type for decimal records, between comma and period.

- **By default, period . is selected as the separator.**

Persistent connection

Set whether connections will be closed after your scripts run in Scriptcase applications.

- **By default, Scriptcase disables this option.**

Use schema before table name

Set whether the database schema is displayed before table names.

- **By default, Scriptcase enables this option.**
  - **Ex:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Interbase connection with PDO Firebird

In Scriptcase, we have the following driver available for Interbase connections: **PDO Firebird.** If you are using your own pre-configured environment, Interbase extensions must be enabled manually in PHP.

Prerequisites

If you are using a manual installation on Linux, you will need to install the php extension for Firebird. See below for how to do this.

1 - Log into your **Linux** terminal and type this line below according to your operating system to install the Firebird PDO Firebird driver.

Ubuntu/Debian	CentOS
sudo apt-get install php8.1-interbase	sudo yum install php-interbase

2 - Check if the PDO Firebird driver is enabled in your Scriptcase diagnostic. Here is how to locate your diagnostics and verify that the driver is enabled.

- When you access the top menu **Help > Diagnostics**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a connection in Scriptcase

Here's how to create a connection in your Scriptcase project using the Firebird database.

1 - Access any project from your Scriptcase.

2 - Click on the **New connection** icon to create a connection.

or go to the **Database > New Connection** menu.

After that a new page will appear with all the database connections.

3 - Select the **Firebird** connection.

Connection

Enter the parameters for connecting to your Firebird database as follows:

- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix**conn** along with the database name.
- **DBMS Driver**: Select the Firebird Driver used to connect. In this example we use the**Firebird PDO** Driver.
- **Server/Host (Name or IP):** Enter the IP or domain of the server where the database is installed.
  - **Ex:** serverdomain.com or 192.168.254.170.
- **Port**: Enter the numeric port for your database server.
  - **Ex:** 3050
- **Database Name**: Enter the database created for use by the entered user.
- **Username**: Enter the user to authenticate the connection to your Firebird database.

- **Password:** Enter the password to complete the authentication process.
- **Test Connection**: Click this button to get a response from Scriptcase request to know if the entered parameters are correct.

Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.

Show

Allows the Firebird connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option will display your database views.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables are displayed. The setting can contain a PREFIX% or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

- **Owner:** Inform the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above.**
- **Show:** Choose whether tables are displayed for the owner entered.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to connection-specific settings. Changes made in this section impact the data display and performance of the application.

Charset

Select the encoding used in your database.

Role

Defines the user role for the database session. Roles are used to manage permissions and access control

within Firebird.

You can set more than one role for the user separating them by using the “;”

Dialect

Determines the SQL dialect used in the connection. Firebird supports different dialects (1, 2, or 3), which affect SQL syntax and behavior.

If the option in the database is set to 2, it doesn’t matter what option you choose in this field.

- **Options**:
  - **Empty** - Gets the default from the database.
  - **1** - Old version of Firebird (1.5) > Does not support timestamp, Numeric or Decimal
  - **2** - Version with timestamp support
  - **3** - Version - Newer versions (Case sensitive and compatible with Firebird 2.5, 3 and 4)

Decimal Separator

Select the separator type for decimal records, between comma and period.

- **By default, period . is selected as the separator.**

Persistent connection

Set whether connections will be closed after your scripts run in Scriptcase applications.

- **By default, Scriptcase disables this option.**

Use schema before table name

Set whether the database schema is displayed before table names.

- **By default, Scriptcase enables this option.**
  - **Ex:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will

appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Interbase connection with PDO Firebird

In Scriptcase, we have the following drivers available for Firebird connections: **Firebird PDO, Firebird.** If you are using your own pre-configured environment, the Firebird extensions must be enabled manually in PHP.

Prerequisites

If you are using a manual installation on MacOS, you will need to install PHP and the Firebird driver will be enabled. Click here to see how to do this.

1 - Check that the Firebird driver is enabled in your Scriptcase diagnostics. Here is how to locate your diagnostics below and verify that the driver is enabled.

- By going to the top menu **Help > Diagnostics**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a connection in Scriptcase

Here's how to create a connection in your Scriptcase project using the Firebird database.

1 - Access any project from your Scriptcase.

2 - Click on the **New connection** icon to create a connection.

or go to the **Database > New Connection** menu.

After that a new page will appear with all the database connections.

3 - Select the **Firebird** connection.

Connection

Enter the parameters for connecting to your Firebird database as follows:

- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Firebird Driver used to connect. In this example we use the **Firebird PDO** Driver.
- **Server/Host (Name or IP):** Enter the IP or domain of the server where the database is installed.
  - **Ex:** serverdomain.com or 192.168.254.170.
- **Port**: Enter the numeric port for your database server.
  - **Ex:** 3050
- **Database Name**: Enter the database created for use by the entered user.
- **Username**: Enter the user to authenticate the connection to your Firebird database.
- **Password:** Enter the password to complete the authentication process.
- **Test Connection**: Click this button to get a response from Scriptcase request to know if the entered parameters are correct.

Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.

Show

Allows the Firebird connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option will display your database views.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables are displayed. The setting can contain a PREFIX% or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

- **Owner:** Inform the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above.**
- **Show:** Choose whether tables are displayed for the owner entered.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to connection-specific settings. Changes made in this section impact the data display and performance of the application.

Charset

Select the encoding used in your database.

Role

Defines the user role for the database session. Roles are used to manage permissions and access control within Firebird.

You can set more than one role for the user separating them by using the “;”

Dialect

Determines the SQL dialect used in the connection. Firebird supports different dialects (1, 2, or 3), which affect SQL syntax and behavior.

If the option in the database is set to 2, it doesn’t matter what option you choose in this field.

- **Options**:
  - **Empty** - Gets the default from the database.
  - **1** - Old version of Firebird (1.5) > Does not support timestamp, Numeric or Decimal
  - **2** - Version with timestamp support
  - **3** - Version - Newer versions (Case sensitive and compatible with Firebird 2.5, 3 and 4)

Decimal Separator

Select the separator type for decimal records, between comma and period.

- **By default, period . is selected as the separator.**

Persistent connection

Set whether connections will be closed after your scripts run in Scriptcase applications.

- **By default, Scriptcase disables this option.**

Use schema before table name

Set whether the database schema is displayed before table names.

- **By default, Scriptcase enables this option.**
  - **Ex:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

MySQL with Scriptcase

In Scriptcase, we have two drivers to connecting with MySQL. You must choose the most suitable for your environment. It is recommended to use **MySQL PDO** which is more faster, however it is up to you to choose the driver. **If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP**.

You can check your enabled drivers by accessing your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

If the desired driver is already enabled and you just want to know how to connect with Scriptcase, click on the link below according to the driver that will be used.

Environment	MySQL PDO	MySQLi
Windows	See how to connect	See how to connect
Linux	See how to connect	See how to connect
macOS	See how to connect	See how to connect

Connecting with MySQL PDO

In Scriptcase, we have the following drivers available for connections with MySQL: **MySQL PDO and MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Windows, you'll need to enable the MySQL extensions on **php.ini** file. Check below how to do it.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the MySQL extensions **php_pdo_mysql** and **php_mysqli** removing the ; from the beginning of the line. See the example below:

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 or ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

> **Server sent charset unknown to the client** If you are using MySQL 8, and your Scriptcase is using PHP 7.0 or 7.3, due to the PHP 7 bug, an error occurs when trying to connect to the Scriptcase. Click here and see how to work around this error.

See below how to create a connection in your Scriptcase project using the MySQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **MySQL** connection.

Connection

Enter the parameters for connecting to your MySQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MySQL Driver used to connect. In this example, we use the **MySQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Username**: Inform the user to authenticate the connection to your MySQL database.
- **Password**: Enter the password to complete the authentication process.

- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new MySQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

On this tab you can set your SSL encryption for your database connection. Check below how to do this.

Use SSL

Activates or deactivates the use of secure connections with MySQL.

Client Key

Path to the client's private identification key in the format PEM.

Client Certificate

Path to the client's public certificate key.

CA Path

Path to the directory that contains the Certificate of Authority (CA) in the format PEM, if used, it should specify the same certificate used by the server.

CA Certificate

Path to the Certificate of Authority (AC) in the format PEM. This option, if used, should specify the same certificate as the server.

Specific Chipper

A list of permitted numbers to use with the cryptography of the connection. If one of the numbers is not supported, the SSL connection won't work.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the MySQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with MySQLi

In Scriptcase, we have the following drivers available for connections with MySQL: **MySQL PDO and MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Windows, you'll need to enable the MySQL extensions on **php.ini** file. Check below how to do it.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the MySQL extensions **php_pdo_mysql** and **php_mysqli** removing the ; from the beginning of the line. See the example below:

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 or ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

> **Server sent charset unknown to the client** If you are using MySQL 8, and your Scriptcase is using PHP 7.0 or 7.3, due to the PHP 7 bug, an error occurs when trying to connect to the Scriptcase. Click here and see how to work around this error.

See below how to create a connection in your Scriptcase project using the MySQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **MySQL** connection.

Connection

Enter the parameters for connecting to your MySQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MySQL Driver used to connect. In this example, we use the **MySQLi** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Username**: Inform the user to authenticate the connection to your MySQL database.
- **Password**: Enter the password to complete the authentication process.

- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new MySQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

On this tab you can set your SSL encryption for your database connection. Check below how to do this.

Use SSL

Activates or deactivates the use of secure connections with MySQL.

Client Key

Path to the client's private identification key in the format PEM.

Client Certificate

Path to the client's public certificate key.

CA Path

Path to the directory that contains the Certificate of Authority (CA) in the format PEM, if used, it should specify the same certificate used by the server.

CA Certificate

Path to the Certificate of Authority (AC) in the format PEM. This option, if used, should specify the same certificate as the server.

Specific Chipper

A list of permitted numbers to use with the cryptography of the connection. If one of the numbers is not supported, the SSL connection won't work.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the MySQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with MySQL PDO

In Scriptcase, we have the following drivers available for connections with MySQL: **MySQL PDO and MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Linux, you'll need to install the MySQL php extension. Check below how to do it.

1 - Access your **Linux** terminal and type this line below according to your OS to install MySQL PDO driver.

Ubuntu	CentOS
sudo apt-get install php8.1-mysql	sudo yum install php-mysql

2 - Check if the MySQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection with ScriptCase

> **Server sent charset unknown to the client** If you are using MySQL 8, and your Scriptcase is using PHP 7.0 or 7.3, due to the PHP 7 bug, an error occurs when trying to connect to the Scriptcase. Click here and see how to work around this error.

See below how to create a connection in your Scriptcase project using the MySQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **MySQL** connection.

Connection

Enter the parameters for connecting to your MySQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MySQL Driver used to connect. In this example, we use the **MySQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.

- **Username**: Inform the user to authenticate the connection to your MySQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new MySQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

On this tab you can set your SSL encryption for your database connection. Check below how to do this.

Use SSL

Activates or deactivates the use of secure connections with MySQL.

Client Key

Path to the client’s private identification key in the format PEM.

Client Certificate

Path to the client’s public certificate key.

CA Path

Path to the directory that contains the Certificate of Authority (CA) in the format PEM, if used, it should specify the same certificate used by the server.

CA Certificate

Path to the Certificate of Authority (AC) in the format PEM. This option, if used, should specify the same certificate as the server.

Specific Chipper

A list of permitted numbers to use with the cryptography of the connection. If one of the numbers is not supported, the SSL connection won’t work.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the MySQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with MySQLi

In Scriptcase, we have the following drivers available for connections with MySQL: **MySQL PDO and MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Linux, you'll need to install the MySQL php extension. Check below how to do it.

1 - Access your **Linux** terminal and type this line below according to your OS to install MySQLi driver.

Ubuntu	CentOS
sudo apt-get install php8.1-mysql	sudo yum install php-mysql

2 - Check if the MySQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection with ScriptCase

> **Server sent charset unknown to the client** If you are using MySQL 8, and your Scriptcase is using PHP 7.0 or 7.3, due to the PHP 7 bug, an error occurs when trying to connect to the Scriptcase. Click here and see how to work around this error.

See below how to create a connection in your Scriptcase project using the MySQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **MySQL** connection.

Connection

Enter the parameters for connecting to your MySQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MySQL Driver used to connect. In this example, we use the **MySQLi** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.

- **Username**: Inform the user to authenticate the connection to your MySQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new MySQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

On this tab you can set your SSL encryption for your database connection. Check below how to do this.

Use SSL

Activates or deactivates the use of secure connections with MySQL.

Client Key

Path to the client’s private identification key in the format PEM.

Client Certificate

Path to the client’s public certificate key.

CA Path

Path to the directory that contains the Certificate of Authority (CA) in the format PEM, if used, it should specify the same certificate used by the server.

CA Certificate

Path to the Certificate of Authority (AC) in the format PEM. This option, if used, should specify the same certificate as the server.

Specific Chipper

A list of permitted numbers to use with the cryptography of the connection. If one of the numbers is not supported, the SSL connection won’t work.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the MySQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with MySQL PDO

In Scriptcase, we have the following drivers available for connections with MySQL: **MySQL PDO and MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually installed with PHP.

Prerequisites

If you are using a manual installation on MacOS, you'll need to install the PHP and the MySQL driver will be enabled. Click here to see how to do it.

1 - Check if the MySQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection with ScriptCase

> **Server sent charset unknown to the client** If you are using MySQL 8, and your Scriptcase is using PHP 7.0 or 7.3, due to the PHP 7 bug, an error occurs when trying to connect to the Scriptcase. Click here and see how to work around this error.

See below how to create a connection in your Scriptcase project using the MySQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **MySQL** connection.

Connection

Enter the parameters for connecting to your MySQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MySQL Driver used to connect. In this example, we use the **MySQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Username**: Inform the user to authenticate the connection to your MySQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.

- **Create Database:** Clicking on this button, you will create a new MySQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

On this tab you can set your SSL encryption for your database connection. Check below how to do this.

Use SSL

Activates or deactivates the use of secure connections with MySQL.

Client Key

Path to the client’s private identification key in the format PEM.

Client Certificate

Path to the client’s public certificate key.

CA Path

Path to the directory that contains the Certificate of Authority (CA) in the format PEM, if used, it should specify the same certificate used by the server.

CA Certificate

Path to the Certificate of Authority (AC) in the format PEM. This option, if used, should specify the same certificate as the server.

Specific Chipper

A list of permitted numbers to use with the cryptography of the connection. If one of the numbers is not supported, the SSL connection won’t work.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the MySQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with MySQLi

In Scriptcase, we have the following drivers available for connections with MySQL: **MySQL PDO and MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually installed with PHP.

Prerequisites

If you are using a manual installation on MacOS, you'll need to install the PHP and the MySQL driver will be enabled. Click here to see how to do it.

1 - Check if the MySQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection with ScriptCase

> **Server sent charset unknown to the client** If you are using MySQL 8, and your Scriptcase is using PHP 7.0 or 7.3, due to the PHP 7 bug, an error occurs when trying to connect to the Scriptcase. Click here and see how to work around this error.

See below how to create a connection in your Scriptcase project using the MySQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **MySQL** connection.

Connection

Enter the parameters for connecting to your MySQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MySQL Driver used to connect. In this example, we use the **MySQLi** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Username**: Inform the user to authenticate the connection to your MySQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new MySQL database for use.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

On this tab you can set your SSL encryption for your database connection. Check below how to do this.

Use SSL

Activates or deactivates the use of secure connections with MySQL.

Client Key

Path to the client's private identification key in the format PEM.

Client Certificate

Path to the client's public certificate key.

CA Path

Path to the directory that contains the Certificate of Authority (CA) in the format PEM, if used, it should specify the same certificate used by the server.

CA Certificate

Path to the Certificate of Authority (AC) in the format PEM. This option, if used, should specify the same certificate as the server.

Specific Chipper

A list of permitted numbers to use with the cryptography of the connection. If one of the numbers is not supported, the SSL connection won't work.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the MySQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Oracle with Scriptcase

In Scriptcase, we have several drivers for connecting to Oracle. You must choose the most suitable for your environment. It is recommended to use **Oracle PDO** which is more faster, however it is up to you to choose the driver. **If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP**.

You can check your enabled drivers by accessing your diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

Automatic Installation

If the desired driver is already enabled and you just want to know how to connect with Scriptcase, click on the link below according to the driver that will be used.

Environment	Oracle PDO	Oracle 8.0.5 ou Acima	Oracle ODBC	Oracle 8
Windows	See how to connect	See how to connect	See how to connect	See how to connect
Linux	See how to connect	See how to connect	See how to connect	See how to connect
macOS	See how to connect	See how to connect	See how to connect	See how to connect

Manual Installation

If the driver you want to use is disabled, see how to enable each one. Click on the link below according to the driver to be used.

Environment	Oracle PDO	Oracle 8.0.5 ou Acima	Oracle ODBC	Oracle 8
Windows	See how to connect	See how to connect	See how to connect	See how to connect
Linux	See how to connect	See how to connect	See how to connect	See how to connect
macOS	See how to connect	See how to connect	See how to connect	See how to connect

Connecting with Oracle PDO

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the installer downloaded. **Download the Oracle Instant Client according to your PHP architecture.**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed the tool using Scripcase installer, it is not necessary to download the **Microsoft Visual C++ 2017**. For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017.**

Required Files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click Here
- Microsoft Visual C++ **2017(x64)**: Click Here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click Here
- Microsoft Visual C++ **2017(x86)**: Click Here

  WARNING: PHP 8.1 is compatible with **Oracle Instant Client 11 or Higher**.
  To download Oracle Instant Client lower than version **19.6.0.0**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle PDO on Windows

Scriptcase's automatic installation comes with pre-configured Oracle drivers, just missing **Instant_Client** to be configured to make the connection. Follow the steps described below to enable the connection drivers.

1 - Extract the previously downloaded **Instant Client Basic** package to your computer's root folder.

E.g.,

C:\instantclient_19_6

2 - Paste the DLLs listed below into the Scriptcase **PHP** folder C:\Program Files\NetMake\v9-php81\components\php:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add in the **Path** parameter of the Environment Variables the full path to the **Instant Client**. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced System Settings.**

- Click on the **Advanced** > **Environment Variables** tab.

- Under **System Variables**, Select the **Path** item and click **Edit**.

- Click the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.

- **E.g.** serverdomain.com/XE or 192.168.254.170/XE
- For more information about Oracle's **Service Name**, click here.

- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8.0.5 or Higher

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the installer downloaded. **Download the Oracle Instant Client according to your PHP architecture.**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed the tool using Scripcase installer, it is not necessary to download the **Microsoft Visual C++ 2017**. For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017.**

Required Files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click Here
- Microsoft Visual C++ **2017(x64)**: Click Here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click Here
- Microsoft Visual C++ **2017(x86)**: Click Here

  WARNING: PHP 8.1 is compatible with **Oracle Instant Client 11 or Higher**.
  To download Oracle Instant Client lower than version **19.6.0.0**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click Here

Oracle 8.0.5 or Higher Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8.0.5 or Higher on Windows

Scriptcase's automatic installation comes with pre-configured Oracle drivers, just missing **Instant_Client** to be configured to make the connection. Follow the steps described below to enable the connection drivers.

1 - Extract the previously downloaded **Instant Client Basic** package to your computer's root folder.

E.g.,

C:\instantclient_19_6

2 - Paste the DLLs listed below into the Scriptcase **PHP** folder C:\Program Files\NetMake\v9-php81\components\php:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add in the **Path** parameter of the Environment Variables the full path to the **Instant Client**. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced System Settings.**

- Click on the **Advanced** > **Environment Variables** tab.

- Under **System Variables**, Select the **Path** item and click **Edit**.

- Click the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8.0.5 or Higher** Driver.

- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.

- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact

the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the

database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting withte Oracle ODBC

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the installer downloaded. **Download the Oracle Instant Client according to your PHP architecture.**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed the tool using Scripcase installer, it is not necessary to download the **Microsoft Visual C++ 2017**. For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017.**

Required Files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click Here
- Oracle ODBC Package **19.6.0.0(x64)**: Click Here
- Microsoft Visual C++ **2017(x64)**: Click Here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click Here
- Oracle ODBC Package **19.6.0.0(x86)**: Click Here
- Microsoft Visual C++ **2017(x86)**: Click Here

  WARNING: PHP 8.1 is compatible with **Oracle Instant Client 11 or Higher**.
  To download Oracle Instant Client lower than version **19.6.0.0**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Below**: Click Here

Oracle ODBC Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle ODBC on Windows

Scriptcase's automatic installation comes with pre-configured Oracle drivers, just missing **instant_client** and the **ODBC** package to be configured to make the connection. Follow the steps described below to enable the

connection drivers.

1 - Extract the previously downloaded **Instant Client Basic** package to your computer's root folder.

E.g., C:\instantclient_19_6

2 - Extract the previously downloaded **ODBC** package to the Instant Client folder on the root path of your computer.

3 - Run the **odbc_install.exe** file with administrator privileges for the driver to be installed on your system.

4 - Paste the DLLs listed below into the Scriptcase **PHP** folder C:\Program Files\NetMake\v9-php81\components\php:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

5 - Add in the **Path** parameter of the Environment Variables the full path to the **Instant Client**. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced System Settings.**

- Click on the **Advanced** > **Environment Variables** tab.

- Under **System Variables**, Select the **Path** item and click **Edit**.

- Click the **New** button and add the path to the instant client as shown below:

6 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a System ODBC Data Source

Check below the entire step by step to create an ODBC data source to connect to Scriptcase.

> **NOTE**: ODBC must be created on the same server where ScriptCase is installed.

1 - Access your **ODBC Data Source Manager** and select according to your architecture. We will be selecting the **x64** version:

2 - When accessing the ODBC Data Sources Administrator, select the **DSN System** tab and click on **Add** to create your connection to your Oracle Database.

3 - After that, it is necessary to select the Driver to connect to Oracle. Select the Driver: **Oracle in instantclient_19_5**

4 - Now, enter the connection parameters as shown below:

- **Data Source Name**: Define here the name of the ODBC object that will be used when connecting to Scriptcase.
- **Description**: Write a definition for the Data Source, e.g., what is the purpose?.
- **TNS Service Name**: Enter the server's IP or domain here followed by the Oracle SERVICE NAME.

- **User ID**: Inform the user that will be used in the connection for authentication.

5 - Test the connection to check if all parameters have been filled in correctly.

You will be asked to enter the password to connect.

6 - Click **OK** to complete this process.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle ODBC** Driver.
- **Server/Host (Name or IP)**: Enter the Data Source Name created in your ODBC manager.
  - **E.G.** oracle
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the installer downloaded. **Download the Oracle Instant Client according to your PHP architecture.**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed the tool using Scripcase installer, it is not necessary to download the **Microsoft Visual C++ 2017**. For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017.**

Required Files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click Here
- Microsoft Visual C++ **2017(x64)**: Click Here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click Here
- Microsoft Visual C++ **2017(x86)**: Click Here

  WARNING: PHP 8.1 is compatible with **Oracle Instant Client 11 or Higher**.
  To download Oracle Instant Client lower than version **19.6.0.0**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click Here

Oracle 8 Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8 on Windows

Scriptcase's automatic installation comes with pre-configured Oracle drivers, just missing **Instant_Client** to be configured to make the connection. Follow the steps described below to enable the connection drivers.

1 - Extract the previously downloaded **Instant Client Basic** package to your computer's root folder.

E.g.,

C:\instantclient_19_6

2 - Paste the DLLs listed below into the Scriptcase **PHP** folder C:\Program Files\NetMake\v9-php81\components\php:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add in the **Path** parameter of the Environment Variables the full path to the **Instant Client**. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced System Settings.**

- Click on the **Advanced** > **Environment Variables** tab.

- Under **System Variables**, Select the **Path** item and click **Edit**.

- Click the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along

with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.

- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle PDO

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  WARNING: Download the Oracle instant client according to your PHP architecture.

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. **For manual installations, you must download and install both versions of Microsoft Visual C ++ 2017.**

Required Files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click Here
- Microsoft Visual C++ **2017(x64)**: Click Here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click Here
- Microsoft Visual C++ **2017(x86)**: Click Here

  WARNING: PHP 7.3 is compatible with **Oracle Instant Client 11 or Higher**.
  To download Oracle Instant Client lower than version **19.6.0.0**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle PDO on Windows

It is necessary to enable the extension in PHP and configure **instant_client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the oracle extensions **oci8_12c** and **pdo_oci** removing the ; from the beginning of the line. See the example below:

2 - Extract the previously downloaded **Instant Client Basic** package to your computer's root folder.

E.g.,

C:\instantclient_19_6

3 - Paste the DLLs listed below into the Scriptcase **PHP** folder C:\php:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

4 - Add in the **Path** parameter of the Environment Variables the full path to the **Instant Client**. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced System Settings.**

- Click on the **Advanced** > **Environment Variables** tab.

- Under **System Variables**, Select the **Path** item and click **Edit**.

- Click the **New** button and add the path to the instant client as shown below:

5 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle PDO** Driver.

- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.

- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact

the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the

database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8.0.5 or Higher

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  WARNING: Download the Oracle instant client according to your PHP architecture.

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. **For manual installations, you must download and install both versions of Microsoft Visual C ++ 2017.**

Required Files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click Here
- Microsoft Visual C++ **2017(x64)**: Click Here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click Here
- Microsoft Visual C++ **2017(x86)**: Click Here

  WARNING: PHP 7.3 is compatible with **Oracle Instant Client 11 or Higher**.
  To download Oracle Instant Client lower than version **19.6.0.0**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click Here

Oracle 8.0.5 or Higher Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8.0.5 or Higher on Windows

It is necessary to enable the extension in PHP and configure **instant_client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the oracle extensions **oci8_12c** and **pdo_oci** removing the ; from the beginning of the line. See the example below:

2 - Extract the previously downloaded **Instant Client Basic** package to your computer's root folder.

E.g.,

C:\instantclient_19_6

3 - Paste the DLLs listed below into the Scriptcase **PHP** folder C:\php:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

4 - Add in the **Path** parameter of the Environment Variables the full path to the **Instant Client**. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced System Settings.**

- Click on the **Advanced** > **Environment Variables** tab.

- Under **System Variables**, Select the **Path** item and click **Edit**.

- Click the **New** button and add the path to the instant client as shown below:

5 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8.0.5 or**

Higher Driver.

- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.

- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle ODBC

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  WARNING: Download the Oracle instant client according to your PHP architecture.

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

For manual installations, you must download and install both versions of Microsoft Visual C ++ 2017.

Required Files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click Here
- Oracle ODBC Package **19.6.0.0(x64)**: Click Here
- Microsoft Visual C++ **2017(x64)**: Click Here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click Here
- Oracle ODBC Package **19.6.0.0(x86)**: Click Here
- Microsoft Visual C++ **2017(x86)**: Click Here

  WARNING: PHP 7.3 is compatible with **Oracle Instant Client 11 or Higher**.
  To download Oracle Instant Client lower than version **19.6.0.0**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click Here

Oracle ODBC Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle ODBC on Windows

It is necessary to enable the extension in PHP and configure **instant_client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the oracle extensions **oci8_12c** and **pdo_oci** removing the ; from the beginning of the line. See the example below:

2 - Extract the previously downloaded **Instant Client Basic** package to your computer's root folder.

E.g., C:\instantclient_19_6

3 - Paste the DLLs listed below into the Scriptcase **PHP** folder C:\php:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

4 - Add in the **Path** parameter of the Environment Variables the full path to the **Instant Client**. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced System Settings.**

- Click on the **Advanced** > **Environment Variables** tab.

- Under **System Variables**, Select the **Path** item and click **Edit**.

- Click the **New** button and add the path to the instant client as shown below:

5 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 service and right click on this service, then **Restart**.

Creating a System ODBC Data Source

Check below the entire step by step to create an ODBC data source to connect to Scriptcase.

> **NOTE**: ODBC must be created on the same server where ScriptCase is installed.

1 - Access your **ODBC Data Source Manager** and select according to your architecture. We will be selecting the **x64** version:

2 - When accessing the ODBC Data Sources Administrator, select the **DSN System** tab and click on **Add** to create your connection to your Oracle Database.

3 - After that, it is necessary to select the Driver to connect to Oracle. Select the Driver: **Oracle in instantclient_19_5**

4 - Now, enter the connection parameters as shown below:

- **Data Source Name**: Define here the name of the ODBC object that will be used when connecting to Scriptcase.
- **Description**: Write a definition for the Data Source, e.g., what is the purpose?.
- **TNS Service Name**: Enter the server's IP or domain here followed by the Oracle SERVICE NAME.
- **User ID**: Inform the user that will be used in the connection for authentication.

5 - Test the connection to check if all parameters have been filled in correctly.

You will be asked to enter the password to connect.

6 - Click **OK** to complete this process.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle ODBC** Driver.
- **Server/Host (Name or IP)**: Enter the Data Source Name created in your ODBC manager.
  - **E.G.** oracle
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items

selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  WARNING: Download the Oracle instant client according to your PHP architecture.

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. **For manual installations, you must download and install both versions of Microsoft Visual C ++ 2017.**

Required Files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click Here
- Microsoft Visual C++ **2017(x64)**: Click Here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click Here
- Microsoft Visual C++ **2017(x86)**: Click Here

  WARNING: PHP 7.3 is compatible with **Oracle Instant Client 11 or Higher**.
  To download Oracle Instant Client lower than version **19.6.0.0**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click Here

Oracle 8 Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8 on Windows

It is necessary to enable the extension in PHP and configure **instant_client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the oracle extensions **oci8_12c** and **pdo_oci** removing the ; from the beginning of the line. See the example below:

2 - Extract the previously downloaded **Instant Client Basic** package to your computer's root folder.

E.g.,

C:\instantclient_19_6

3 - Paste the DLLs listed below into the Scriptcase **PHP** folder C:\php:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

4 - Add in the **Path** parameter of the Environment Variables the full path to the **Instant Client**. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced System Settings.**

- Click on the **Advanced** > **Environment Variables** tab.

- Under **System Variables**, Select the **Path** item and click **Edit**.

- Click the **New** button and add the path to the instant client as shown below:

5 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8** Driver.

- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.

- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact

the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the

database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle PDO

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

> **IMPORTANT:** If you are using the automatic installer, **the extensions are already configured in PHP**, ready to connect if you have enabled them in the installation process. If you have not enabled them, see below how to enable them.

Configuring Oracle PDO on Linux (Automatic installation)

The automatic installation of Scriptcase already comes with the Oracle drivers pre-configured, with only the **instant_client** to be configured to make the connection. Follow the steps described below to enable connection drivers.

1 - Download the Oracle Instant Client **12.1.0.2** by clicking here.

2 - After downloading, extract the file and copy the folder **instantclient** into the directory: **/opt/Scriptcase/v9-php81/components/drivers/**

3 - Restart the Apache sudo service apachesc9php81 restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8.0.5 or Higher

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

> **IMPORTANT:** If you are using the automatic installer, **the extensions are already configured in PHP**, ready to connect if you have enabled them in the installation process. If you have not enabled them, see below how to enable them.

Configuring Oracle 8.0.5 or Higher on Linux (Automatic installation)

The automatic installation of Scriptcase already comes with the Oracle drivers pre-configured, with only the **instant_client** to be configured to make the connection. Follow the steps described below to enable connection drivers.

1 - Download the Oracle Instant Client **12.1.0.2** by clicking here.

2 - After downloading, extract the file and copy the folder **instantclient** into the directory: **/opt/Scriptcase/v9-php81/components/drivers/**

3 - Restart the Apache sudo service apachesc9php81 restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8.0.5 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle ODBC

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the installer downloaded. **Download the Oracle Instant Client according to your PHP architecture.**

- **i386/i686** Architecture = **32 bits**
- **x86_64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required Files:

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click Here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click Here
- Oracle Instant Client - ODBC Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click Here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click Here
- Oracle Instant Client - ODBC Package **12.1.0.2(i386/i686)**: Click here

WARNING: PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle ODBC Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle ODBC on Linux

Scriptcase's automatic installation comes with pre-configured Oracle drivers, just missing **Instant_Client** to be configured to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

E.g.

> **NOTE:** If you are using CentOS/RHEL, it is not necessary to use 'export' before the variable.

4 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apachesc9php73 restart	sudo systemctl restart apachesc9php73

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle ODBC** Driver.
- **Server/Host (Name or IP)**: Enter the Data Source Name created in your ODBC manager.
  - **E.G.** oracle
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**

- **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

> **IMPORTANT:** If you are using the automatic installer, **the extensions are already configured in PHP**, ready to connect if you have enabled them in the installation process. If you have not enabled them, see below how to enable them.

Configuring Oracle 8 on Linux (Automatic installation)

The automatic installation of Scriptcase already comes with the Oracle drivers pre-configured, with only the **instant_client** to be configured to make the connection. Follow the steps described below to enable connection drivers.

1 - Download the Oracle Instant Client **12.1.0.2** by clicking here.

2 - After downloading, extract the file and copy the folder **instantclient** into the directory: **/opt/Scriptcase/v9-php81/components/drivers/**

3 - Restart the Apache `sudo service apachesc9php81 restart`

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**
- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your

scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**
    - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle PDO

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  WARNING: Download the Oracle instant client according to your PHP architecture.

- **i386/i686** Architecture = **32 bits**
- **x86_64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required Files:

- **oci8.so** extension: Click Here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

  WARNING: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Oracle database version**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle PDO on Linux

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/

E.g.

NOTE: If you are using CentOS/RHEL, it is not necessary to use ‘export’ before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle’s **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8.0.5 or Higher

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  WARNING: Download the Oracle instant client according to your PHP architecture.

- **i386/i686** Architecture = **32 bits**
- **x86_64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required Files:

- **oci8.so** extension: Click Here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

  WARNING: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle 8.0.5 or Higher Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8 or Highter	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Oracle database version**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8.0.5 or Higher on Linux

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/

E.g.

NOTE: If you are using CentOS/RHEL, it is not necessary to use ‘export’ before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8.0.5 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle’s **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle ODBC

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  WARNING: Download the Oracle instant client according to your PHP architecture.

- **i386/i686** Architecture = **32 bits**
- **x86_64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required Files:

- **oci8.so** extension: Click Here

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click Here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click Here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click Here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click Here

  WARNING: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle ODBC Driver Compatibility Table

Drive	Oracle Database Version						
Oracle ODBC	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Oracle database version**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle ODBC on Linux

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/

E.g.

> **NOTE:** If you are using CentOS/RHEL, it is not necessary to use 'export' before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle ODBC** Driver.
- **Server/Host (Name or IP)**: Enter the Data Source Name created in your ODBC manager.
  - **E.G.** oracle
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. In order for the drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In Scriptcase, there is the **info.php** file, access it through the URL: http://127.0.0.1/scriptcase/info.php, where you will find information regarding the architecture in phpinfo().

  WARNING: Download the Oracle instant client according to your PHP architecture.

- **i386/i686** Architecture = **32 bits**
- **x86_64** Architecture = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required Files:

- **oci8.so** extension: Click Here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

  WARNING: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle 8 Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Instant Client	Oracle database version						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8 on Linux

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/

E.g.

NOTE: If you are using CentOS/RHEL, it is not necessary to use ‘export’ before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle’s **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle PDO

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Click Here

  IMPORTANT: The PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle PDO on MacOs

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** missing to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Restart the Scriptcase apache service for the changes to be saved. Use the command below for this:

EX:

sudo /Applications/Scriptcase/v9-php81/components/apache/bin/apachectl restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8.0.5 or Higher

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Click Here

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle 8.0.5 or Higher Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8.0.5 or Higher on MacOs

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** missing to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the

folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Restart the Scriptcase apache service for the changes to be saved. Use the command below for this:

EX:

sudo /Applications/Scriptcase/v9-php81/components/apache/bin/apachectl restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8.0.5 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle ODBC

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Click Here

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle ODBC Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle ODBC on MacOs

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** missing to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the

folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Restart the Scriptcase apache service for the changes to be saved. Use the command below for this:

EX:

sudo /Applications/Scriptcase/v9-php81/components/apache/bin/apachectl restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle ODBC** Driver.
- **Server/Host (Name or IP)**: Enter the Data Source Name created in your ODBC manager.
  - **E.G.** oracle
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Click Here

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle 8 Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8 on MacOs

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** missing to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the

folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Restart the Scriptcase apache service for the changes to be saved. Use the command below for this:

EX:

sudo /Applications/Scriptcase/v9-php81/components/apache/bin/apachectl restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle PDO

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle PDO on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/8.1/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.

- **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.

- **By default, this option is left blank by Scriptcase.**

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**

- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**

- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8.0.5 or Higher

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle 8.0.5 or Higher Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8.0.5 or Higher on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/8.1/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8.0.5 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.

- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**
- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the

client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**

- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**

- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the

database service’s listening port.

Connecting with Oracle ODBC

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle ODBC Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle ODBC on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/8.1/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle ODBC** Driver.
- **Server/Host (Name or IP)**: Enter the Data Source Name created in your ODBC manager.
  - **E.G.** oracle
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with Oracle 8

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.**

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle 8 Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5 or Higher	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versión de base de datos Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8 on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/8.1/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.

- **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .

- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**

- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.

- **By default, this option is left blank by Scriptcase.**

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**

- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**

- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

PostgreSQL with Scriptcase

In Scriptcase, we have four drivers to connect with PostgreSQL. You must choose the most suitable for your environment.

It is recommended to use PostgreSQL PDO which is faster, however, it is up to you to choose the driver. If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Check below the PostgreSQL versions that you can use in the Scriptcase 9

PostgreSQL versions list supported in Scriptcase 9

PostgreSQL Version	PostgreSQL PDO (**recommended**)	PostgreSQL 7 or Higher	PostgreSQL 6.4 or Higher	PostegreSQL 6.3 or Lower
15	✅	✅	✅	✅
14	✅	✅	✅	✅
13	✅	✅	✅	✅
12	✅	✅	✅	✅
11	✅	✅	✅	✅
10	✅	✅	✅	✅
9.6	✅	✅	✅	✅
9.5	✅	✅	✅	✅
9.4	✅	✅	✅	✅
9.3	✅	✅	✅	✅
9.2	✅	✅	✅	✅
9.1	✅	✅	✅	✅
9.0	✅	✅	✅	✅

Check if PostgreSQL drivers in enable

You can check your enabled drivers by accessing your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

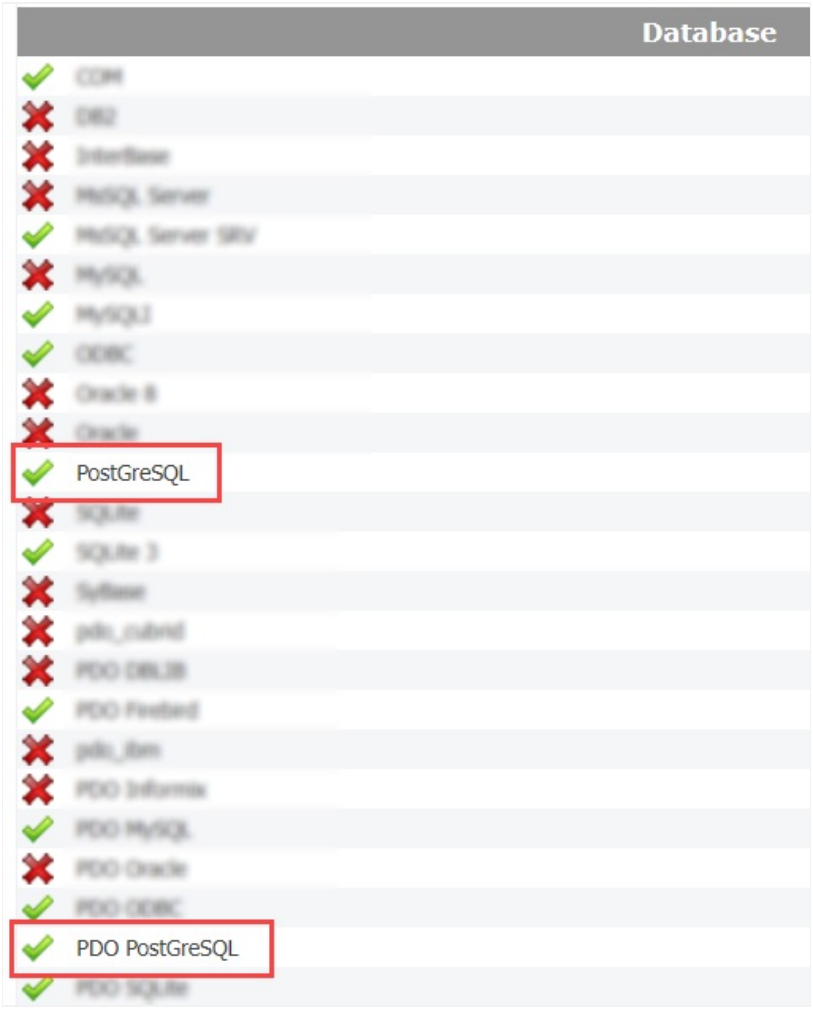
- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

Make sure the driver is enabled, as in the image below:

The extension **PDO PostGreSQL** enable all drivers. The extension **PostGreSQL** enable this drivers: PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.



Click on the link below according to the driver that will be used

Environment	PostgreSQL PDO (**recommended**)	PostgreSQL 7 or Higher	PostgreSQL 6.4 or Higher	PostgreSQL 6.3 or Lower
Windows	Click Here	Click Here	Click Here	Click Here
Linux	Click Here	Click Here	Click Here	Click Here
macOS	Click Here	Click Here	Click Here	Click Here

Connecting with PostgreSQL PDO

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Windows, you'll need to enable the PostgreSQL extensions on **php.ini** file. Check below how to do it.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the PostgreSQL extensions **pdo_pgsql** and **php_pgsql** removing the ; from the beginning of the line. See the example below:

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 or ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.

- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.

- **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL 7 or Higher

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Windows, you'll need to enable the PostgreSQL extensions on **php.ini** file. Check below how to do it.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the PostgreSQL extensions **pdo_pgsql** and **php_pgsql** removing the ; from the beginning of the line. See the example below:

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 or ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL 7 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.

- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.

- **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL 6.4 or Higher

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Windows, you'll need to enable the PostgreSQL extensions on **php.ini** file. Check below how to do it.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the PostgreSQL extensions **pdo_pgsql** and **php_pgsql** removing the ; from the beginning of the line. See the example below:

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 or ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL 6.4 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.

- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.

- **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL 6.3 or Lower

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Windows, you'll need to enable the PostgreSQL extensions on **php.ini** file. Check below how to do it.

1 - In the php.ini file, located in C:\php, uncomment the lines referring to the PostgreSQL extensions **pdo_pgsql** and **php_pgsql** removing the ; from the beginning of the line. See the example below:

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 or ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL 6.3 or lower** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.

- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.

- **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL PDO

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Linux, you'll need to install the PostgreSQL php extension. Check below how to do it.

1 - Access your **Linux** terminal and type this line below according to your OS to install PostgreSQL PDO driver.

Ubuntu	CentOS
sudo apt-get install php8.1-pgsql	sudo yum install php-pgsql

2 - Check if the PostgreSQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL 7 or Higher

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Linux, you'll need to install the PostgreSQL php extension. Check below how to do it.

1 - Access your **Linux** terminal and type this line below according to your OS to install PostgreSQL PDO driver.

Ubuntu	CentOS
sudo apt-get install php8.1-pgsql	sudo yum install php-pgsql

2 - Check if the PostgreSQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL 7 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL 6.4 or Higher

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Linux, you'll need to install the PostgreSQL php extension. Check below how to do it.

1 - Access your **Linux** terminal and type this line below according to your OS to install PostgreSQL PDO driver.

Ubuntu	CentOS
sudo apt-get install php8.1-pgsql	sudo yum install php-pgsql

2 - Check if the PostgreSQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL 6.4 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL 6.3 or Lower

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Linux, you'll need to install the PostgreSQL php extension. Check below how to do it.

1 - Access your **Linux** terminal and type this line below according to your OS to install PostgreSQL PDO driver.

Ubuntu	CentOS
sudo apt-get install php8.1-pgsql	sudo yum install php-pgsql

2 - Check if the PostgreSQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL 6.3 or lower** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**

- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL PDO

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on MacOS, you'll need to install the PHP and the PostgreSQL driver will be enabled. Click here to see how to do it.

1 - Check if the PostgreSQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

![Accessing diagnosis - Help Menu][diagnosis_sc]

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.

- **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL 7 or Higher

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on MacOS, you'll need to install the PHP and the PostgreSQL driver will be enabled. Click here to see how to do it.

1 - Check if the PostgreSQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

![Accessing diagnosis - Help Menu][diagnosis_sc]

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL 7 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.

- **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL 6.4 or Higher

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on MacOS, you'll need to install the PHP and the PostgreSQL driver will be enabled. Click here to see how to do it.

1 - Check if the PostgreSQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

![Accessing diagnosis - Help Menu][diagnosis_sc]

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL 6.4 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.

- **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with PostgreSQL 6.3 or Lower

In Scriptcase, we have the following drivers available for connections with PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or Higher, PostgreSQL 6.4 or Higher and PostgreSQL 6.3 or Lower.** If you are using your own pre-configured environment, PostgreSQL extensions must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on MacOS, you'll need to install the PHP and the PostgreSQL driver will be enabled. Click here to see how to do it.

1 - Check if the PostgreSQL driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

![Accessing diagnosis - Help Menu][diagnosis_sc]

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the PostgreSQL database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Postgres** connection.

Connection

Enter the parameters for connecting to your PostgreSQL database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver used to connect. In this example, we use the **PostgreSQL 6.3 or lower** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your PostgreSQL database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new PostgreSQL database for use.

- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate)

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key)

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate)

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the PostgreSQL connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.

- **By default, Scriptcase enables this option.**

- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

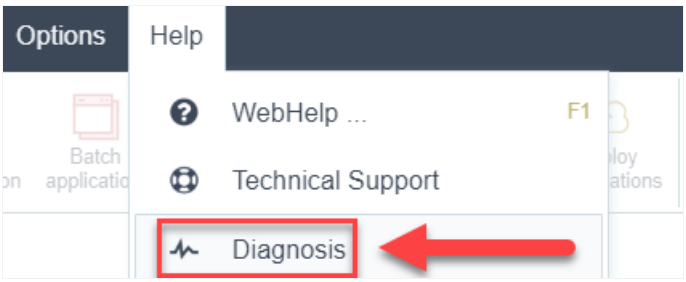
MSSQL Server

Scriptcase provides some driver options for connecting to MSSQL Server, compatible with the latest versions of Windows, Linux and MacOS.

You can see if drivers are enabled by going to diagnosis.

Accessing from the Scriptcase menu.

After login, in the top menu of the tool access Help > Diagnosis.



Accessing directly through the browser

In your browser, without having to login to Scriptcase, you can access the diagnosis.

When installation is local:

`127.0.0.1:8092/scriptcase/diagnosis.php` ou `localhost:8092/scriptcase/diagnosis.php`

In case of installation on a server.

`domain.com/scriptcase/diagnosis.php`

Click on the driver you want to use, according to your operating system.

Windows

- MSSQL Server NATIVE SRV
- MSSQL Server NATIVE SRV PDO
- MSSQL Server ODBC

Linux

- PDO DBLIB

Mac OS

- PDO DBLIB

Compatibility Table

Check below the list of drivers available for each operating system.

OS x Drivers	Native SRV PDO	Native SRV	ODBC	PDO DBlib
Windows Server 2016	✅	✅	✅	❌
Windows Server 2019	✅	✅	✅	❌
Windows Server 2012 /R2	✅	✅	✅	❌
Windows 10	✅	✅	✅	❌
Windows 8 / 8.1	✅	✅	✅	❌
Windows Server 2008 SP2 / R2 SP1	✅	✅	✅	❌
Windows 7 SP1	❌	❌	✅	❌
Windows Vista SP2	❌	❌	✅	❌
Linux	❌	❌	❌	✅
Mac OS	❌	❌	❌	✅

Connection to MSSQL Server NATIVE SRV

At Scriptcase, we have the following drivers available for connecting to SQL Server on Windows: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC .**. If you are using your own PHP, as a pre-configured MSSQL environment, it can be manually configured in PHP.

Requirements

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 13 For SQL Server according to your PHP architecture .**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

File list

Listed below are the necessary files for the drivers to be enabled.

- Download Microsoft ODBC Driver 13 For SQL Server **x64**
- Download Microsoft ODBC Driver 13 For SQL Server **x86**
- SQL Server extensions for PHP

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling MSSQL Server SRV PDO on Windows

If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP** , requiring only the installation of the client. **Proceed to item 3**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64**. See the example below:

```
extension=pdo_sqlsrv_81_nts_x64
extension=sqlsrv_81_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 13 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click Next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart Apache service through **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php81 or Apache2.4 service and right click on that service, then click **Reiniciar**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Microsoft SQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **MSSQL Server** connection.

Connection

Enter the parameters for connecting to your MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the DB2 Driver to connect. In this example, we use the **MSSQL Server NATIVE SRV PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed and its instance separated by a backslash.
- **EX:** dominiodoserver.com\SQLEXPRESS or 192.168.254.170\SQLEXPRESS

- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

Encrypt

This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

trustservercertificate

Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

trustStore

The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

trustStorePassword

The password used to verify the integrity of trustStore data.

If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

hostnameInCertificate

The hostname to use to validate the SQL Server TLS/SSL certificate.

This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to MSSQL Server NATIVE SRV PDO

At Scriptcase, we have the following drivers available for connecting to SQL Server on Windows: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC .**. If you are using your own PHP, as a pre-configured MSSQL environment, it can be manually configured in PHP.

Requirements

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 13 For SQL Server according to your PHP architecture .**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

File list

Listed below are the necessary files for the drivers to be enabled.

- Download Microsoft ODBC Driver 13 For SQL Server **x64**
- Download Microsoft ODBC Driver 13 For SQL Server **x86**
- SQL Server extensions for PHP

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling MSSQL Server SRV PDO on Windows

If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP** , requiring only the installation of the client. **Proceed to item 3**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64**. See the example below:

```
extension=pdo_sqlsrv_81_nts_x64
extension=sqlsrv_81_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 13 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click Next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart Apache service through **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php81 or Apache2.4 service and right click on that service, then click **Reiniciar**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Microsoft SQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **MSSQL Server** connection.

Connection

Enter the parameters for connecting to your MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the DB2 Driver to connect. In this example, we use the **MSSQL Server NATIVE SRV PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed and its instance separated by a backslash.
- **EX:** dominiodoserver.com\SQLEXPRESS or 192.168.254.170\SQLEXPRESS

- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

Encrypt

This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

trustservercertificate

Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

trustStore

The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

trustStorePassword

The password used to verify the integrity of trustStore data.

If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

hostnameInCertificate

The hostname to use to validate the SQL Server TLS/SSL certificate.

This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to MSSQL Server ODBC

At Scriptcase, we have the following drivers available for connecting to SQL Server on Windows: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC .**. If you are using your own PHP, as a pre-configured MSSQL environment, it can be manually configured in PHP.

Requirements

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 13 For SQL Server according to your PHP architecture .**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

File list

Listed below are the necessary files for the drivers to be enabled.

- Download Microsoft ODBC Driver 13 For SQL Server **x64**
- Download Microsoft ODBC Driver 13 For SQL Server **x86**
- SQL Server extensions for PHP

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling MSSQL Server SRV PDO on Windows

If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP** , requiring only the installation of the client. **Proceed to item 3**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64**. See the example below:

```
extension=pdo_sqlsrv_81_nts_x64
extension=sqlsrv_81_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 13 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click Next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart Apache service through **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php81 or Apache2.4 service and right click on that service, then click **Reiniciar**.

Creating the DSN for connection

To connect to the **MSSQL Server ODBC** Driver, you need to configure a System DSN. Follow the steps described below to perform this configuration.

> **ATTENTION**: ODBC must be created on the same Server where ScriptCase is installed.

1 - Access your **ODBC Data Source Manager** and select according to your architecture. We will select the x64 version:

2 - When entering the ODBC Data Sources Administrator, select the **DSN System** tab and click on *_Add* to create your connection to your Database.

3 - After that, it is necessary to select the Driver to connect to the MSSQL Server. Select the Driver: **ODBC Driver 11 for SQL Server .**

4 - Now, you need to define the name of the DSN, the description and the database connection server.

- **Name:** Enter the name that the DSN will have so that you can use it in Scriptcase.
- **Description:** Add a description for the DSN.
- **Server:** inform the database server, port and instance that you will connect to.

4.1 - After entering the data, click on the **Next** button to enter the user name and password required for connection.

- **Login ID:** Inform the user to authenticate with the database.

- **Password:** Enter the corresponding password to authenticate with the informed user.

4.2 - Now, click **Next** to select the default database.

To finish, click **Next** and finally, click **Finish**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Microsoft SQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the menu **Database > New connection**.

After that, a screen will appear with all database connections.

3 - Select the **MSSQL Server** connection.

Connection

Enter the parameters for connecting to your MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the DB2 Driver to connect. In this example, we use the **MSSQL Server ODBC** Driver.
- **Server/Host (Name or IP)**: Enter the system DSN that was configured in ODBC Manager.
- **EX:** mssql
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

Encrypt

This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

trustservercertificate

Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

trustStore

The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

trustStorePassword

The password used to verify the integrity of trustStore data.

If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

hostnameInCertificate

The hostname to use to validate the SQL Server TLS/SSL certificate.

This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to MSSQL Server DBLIB

In Scriptcase installed on Linux, we have the following driver available for connection to MSSQL Server: **DBLIB.** If you are using your own preconfigured environment, the MSSQL Server extensions must be manually enabled in the PHP.

> **IMPORTANT:** if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**“**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Enabling DBLIB on Linux

> **IMPORTANT:** If you are using the Scriptcase automatic installer, **the extensions are already enabled in PHP**, all that is missing is the configuration of the freetds file. **The procedure below is only for manual installations of Scriptcase. To see how to configure freetds in the automatic installation, scroll down to the configuration section by clicking here**

1 - If you do not have the extension enabled you can install the extension with this command in the terminal:

```
sudo apt-get install php8.1-pdo-dblib
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

Install the basic packages using the terminal for this procedure:

Ubuntu\Debian	RHEL\CentOS
sudo apt-get install unixodbc unixodbc-dev gcc nano wget make	sudo yum install unixODBC unixODBC-devel gcc nano wget make

- Download and Unzip the freeTDS:

```
wget https://www.freetds.org/files/stable/freetds-0.95.95.tar.gz

tar -zxf freetds-0.95.95.tar.gz
```

- Access the freeTDS directory and run the following command to compile and install FreeTDS:

```
sudo ./configure --with-tdsver=7.4 --with-unixodbc=/usr --disable-libiconv --disable-static --disable-threadsafe --enable-msdblib --disable-sspi --with-gnu-ld --enable-sybase-compat && make && make install
```

- Edit the FreeTDS configuration file:

Ubuntu\Debian	RHEL\CentOS
sudo nano/usr/local/etc/freetds.conf	sudo vim /usr/local/freetds/freetds.conf

- Add your MSSQL Server connection informations, like this example:

  ```
  [MSSQLServer] host = 192.18.72.03
  port = 1433
  tds version = 7.2
  ```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Setting up freetds - automatic installation

1 - Edit the FreeTDS configuration file:

Automatic installation directory
sudo nano /opt/Scriptcase/v9-php8/components/drivers/freetds/etc/freetds.conf

2 - Add your MSSQL Server connection informations, like in the example:

```
[MSSQLServer] host = 192.18.72.03
port = 1433
tds version = 7.4
```

3 - Restart Apache service

```
sudo service apachesc9php81 restart
```

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver **DBLIB** and the MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **MSSQL Server** connection.

Connection

Enter the parameters for connecting to your MSSQL Server database as follows:

Connection name:

Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.

DBMS Driver

Select the MSSQL Server Driver to connect. In this example, we use the Driver **DBLIB**.

Server/Host (Name or IP)

Enter the name of the data source you have configured on your freetds.conf file.

- **EX:** MSSQLServer
- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

Encrypt

This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

trustservercertificate

Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

trustStore

The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

trustStorePassword

The password used to verify the integrity of trustStore data.

If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

hostnameInCertificate

The hostname to use to validate the SQL Server TLS/SSL certificate.

This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

Decimal Separator:

Select the type of separator for decimal records, between comma and period. * **By default, the . dot is selected as a separator .**

Persistent Connection

Define whether connections will be terminated after the execution of your scripts in Scriptcase applications. * **By default, Scriptcase disables this option .**

Use the schema before the table name

Define whether the database schema will be displayed before the table names. * **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to MSSQL NATIVE SRV PDO

In Scriptcase installed on Linux, we have the following drivers available for connection to MSSQL Server: **DBLIB, NATIVE SRV PDO and NATIVE SRV.** If you are using your own preconfigured environment, the MSSQL Server extensions must be manually enabled in the PHP.

> **IMPORTANT:** if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**“**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Microsoft Driver Installation Tutorial for PHP and SQL Server

> **IMPORTANT:** If you are using Scriptcase automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - Enter the Shell (terminal) as root and install the following packages:

Ubuntu\Debian	RHEL\CentOS
sudo apt-get install unixodbc-dev	sudo yum install unixODBC-devel

In the linux terminal, you will need to run the following commands to install the drivers:

Ubuntu\Debian

```
sudo pecl install pdo_sqlsrv

sudo su
printf "; priority=30\nextension=pdo_sqlsrv.so\n" > /etc/php/8.1/mods-available/pdo_sqlsrv.ini

exit

sudo phpenmod -v 8.1 pdo_sqlsrv
```

RHEL\CentOS

```
sudo pecl install pdo_sqlsrv

sudo su

echo extension=pdo_sqlsrv.so >> `php --ini | grep "Scan for additional .ini files" | sed -e "s|.*:\s*||"`/30-pdo_sqlsrv.ini

exit
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo su

add-apt-repository ppa:ondrej/php -y

sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

> **Important:** Continue to the Microsoft ODBC Driver installation in order to enable the drivers and complete the process.

Installing Microsoft ODBC Driver for SQL Server (Linux)

1 - Using the terminal, run the following commands:

Ubuntu

```
if ! [[ "16.04 18.04 20.04 22.04" == *"$(lsb_release -rs)"* ]];
then
    echo "Ubuntu $(lsb_release -rs) is not currently supported.";
```

```
    exit;
fi

sudo su
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -

curl https://packages.microsoft.com/config/ubuntu/$(lsb_release -rs)/prod.list > /etc/apt/sources.list.d/mssql-release.list

exit

sudo apt-get update

sudo ACCEPT_EULA=Y apt-get install -y msodbcsql17
```

Debian

```
sudo su

curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -

#Download appropriate package for the OS version
#Choose only ONE of the following, corresponding to your OS version

#Debian 9
curl https://packages.microsoft.com/config/debian/9/prod.list > /etc/apt/sources.list.d/mssql-release.list

#Debian 10
curl https://packages.microsoft.com/config/debian/10/prod.list > /etc/apt/sources.list.d/mssql-release.list

#Debian 11
curl https://packages.microsoft.com/config/debian/11/prod.list > /etc/apt/sources.list.d/mssql-release.list

sudo apt-get update

sudo ACCEPT_EULA=Y apt-get install -y msodbcsql17
```

RHEL\CentOS

```
sudo su

#Download appropriate package for the OS version
#Choose only ONE of the following, corresponding to your OS version

#RHEL 7 and Oracle Linux 7
curl https://packages.microsoft.com/config/rhel/7/prod.repo > /etc/yum.repos.d/mssql-release.repo

#RHEL 8 and Oracle Linux 8
curl https://packages.microsoft.com/config/rhel/8/prod.repo > /etc/yum.repos.d/mssql-release.repo

#RHEL 9
curl https://packages.microsoft.com/config/rhel/9.0/prod.repo > /etc/yum.repos.d/mssql-release.repo

sudo yum remove unixODBC-utf16 unixODBC-utf16-devel #to avoid conflicts

sudo ACCEPT_EULA=Y yum install -y msodbcsql17
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Microsoft SQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **MSSQL Server** connection.

Connection

Enter the parameters for connecting to your MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the DB2 Driver to connect. In this example, we use the **MSSQL Server NATIVE SRV PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed and its instance separated by a backslash.
- **EX:** dominiodoserver.com\SQLEXPRESS or 192.168.254.170\SQLEXPRESS
- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

Encrypt

This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

trustservercertificate

Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

trustStore

The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

trustStorePassword

The password used to verify the integrity of trustStore data.

If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

hostnameInCertificate

The hostname to use to validate the SQL Server TLS/SSL certificate.

This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to MSSQL NATIVE SRV

In Scriptcase installed on Linux, we have the following drivers available for connection to MSSQL Server: **DBLIB, NATIVE SRV PDO and NATIVE SRV.** If you are using your own preconfigured environment, the MSSQL Server extensions must be manually enabled in the PHP.

> **IMPORTANT:** if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**“**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Microsoft Driver Installation Tutorial for PHP and SQL Server

> **IMPORTANT:** If you are using Scriptcase automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - Enter the Shell (terminal) as root and install the following packages:

Ubuntu\Debian	RHEL\CentOS
sudo apt-get install unixodbc-dev	sudo yum install unixODBC-devel

In the linux terminal, you will need to run the following commands to install the drivers:

Ubuntu\Debian

```
sudo pecl install sqlsrv

sudo su
printf "; priority=20\nextension=sqlsrv.so\n" > /etc/php/8.1/mods-available/sqlsrv.ini

exit

sudo phpenmod -v 8.1 sqlsrv
```

RHEL\CentOS

```
sudo pecl install sqlsrv

sudo su

echo extension=sqlsrv.so >> `php --ini | grep "Scan for additional .ini files" | sed -e "s|.*:\s*||"`/20-sqlsrv.ini

exit
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo su
add-apt-repository ppa:ondrej/php -y
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

> **Important:** Continue to the Microsoft ODBC Driver installation in order to enable the drivers and complete the process.

Installing Microsoft ODBC Driver for SQL Server (Linux)

1 - Using the terminal, run the following commands:

Ubuntu

```
if ! [[ "16.04 18.04 20.04 22.04" == *"$(lsb_release -rs)"* ]];
then
    echo "Ubuntu $(lsb_release -rs) is not currently supported.";
    exit;
```

```
fi

sudo su
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -

curl https://packages.microsoft.com/config/ubuntu/$(lsb_release -rs)/prod.list > /etc/apt/sources.list.d/mssql-release.list

exit

sudo apt-get update

sudo ACCEPT_EULA=Y apt-get install -y msodbcsql17
```

Debian

```
sudo su

curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -

#Download appropriate package for the OS version
#Choose only ONE of the following, corresponding to your OS version

#Debian 9
curl https://packages.microsoft.com/config/debian/9/prod.list > /etc/apt/sources.list.d/mssql-release.list

#Debian 10
curl https://packages.microsoft.com/config/debian/10/prod.list > /etc/apt/sources.list.d/mssql-release.list

#Debian 11
curl https://packages.microsoft.com/config/debian/11/prod.list > /etc/apt/sources.list.d/mssql-release.list

sudo apt-get update

sudo ACCEPT_EULA=Y apt-get install -y msodbcsql17
```

RHEL\CentOS

```
sudo su

#Download appropriate package for the OS version
#Choose only ONE of the following, corresponding to your OS version

#RHEL 7 and Oracle Linux 7
curl https://packages.microsoft.com/config/rhel/7/prod.repo > /etc/yum.repos.d/mssql-release.repo

#RHEL 8 and Oracle Linux 8
curl https://packages.microsoft.com/config/rhel/8/prod.repo > /etc/yum.repos.d/mssql-release.repo

#RHEL 9
curl https://packages.microsoft.com/config/rhel/9.0/prod.repo > /etc/yum.repos.d/mssql-release.repo

sudo yum remove unixODBC-utf16 unixODBC-utf16-devel #to avoid conflicts

sudo ACCEPT_EULA=Y yum install -y msodbcsql17
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Microsoft SQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **MSSQL Server** connection.

Connection

Enter the parameters for connecting to your MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the DB2 Driver to connect. In this example, we use the **MSSQL Server NATIVE SRV** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed and its instance separated by a backslash.
- **EX:** dominiodoserver.com\SQLEXPRESS or 192.168.254.170\SQLEXPRESS
- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

Encrypt

This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

trustservercertificate

Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

trustStore

The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

trustStorePassword

The password used to verify the integrity of trustStore data.

If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

hostnameInCertificate

The hostname to use to validate the SQL Server TLS/SSL certificate.

This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to PDO DBLIB

In Scriptcase installed on Linux, we have the following driver available for connection to MSSQL Server: **PDO DBLIB .** If you are using your own pre-configured environment, MSSQL Server extensions must be enabled manually in PHP.

> **IMPORTANT:** if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**“**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling the DBLIB PDO on macOS

> **IMPORTANT:** If you are using Scriptcase’s automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

If you have installed PHP 8.1 through our documentation using homebrew, **PDO DBLIB is already enabled in PHP**, requiring only the connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: click here

Connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**MSSQL Server NATIVE SRV or MSSQL Server NATIVE SRV PDO**) and the Azure MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **MSSQL Server** connection.

Connection

Enter the parameters for connecting to your Azure MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MSSQL Server Driver to connect. In this example, we use the Driver **PDO DBLIB**.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed and its instance separated by a backslash.
- **EX:** dominiodoserver.com\SQLEXPRESS or 192.168.254.170\SQLEXPRESS
- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.

- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

SQLite with Scriptcase

In Scriptcase, we only have one driver to connect to SQLite: **SQLite PDO**. **If you are using your own pre-configured environment, the SQLite extension must be enabled manually in PHP**.

You can check if your driver is enabled by accessing your Scriptcase diagnostics. See below for how to locate your diagnostics:

- When you go to the top menu **Help > Diagnostics**, you can easily find it through the interface.

- Or, by accessing the url in your browser:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

If the driver is already enabled, you only need to connect to Scriptcase, click on the link below to be directed to the driver that will be used.

Environment	SQLite PDO
Windows	Connect
Linux	Connect
macOS	Connect

Windows

In Scriptcase, we only have one driver to connect to SQLite: **SQLite PDO**. If you are using your own pre-configured environment, the **SQLite extension must be enabled manually in PHP**.

Prerequisites

If you are using a manual installation on Windows, you will need to enable the SQLite extension in the **php.ini** file. See below for how to do this.

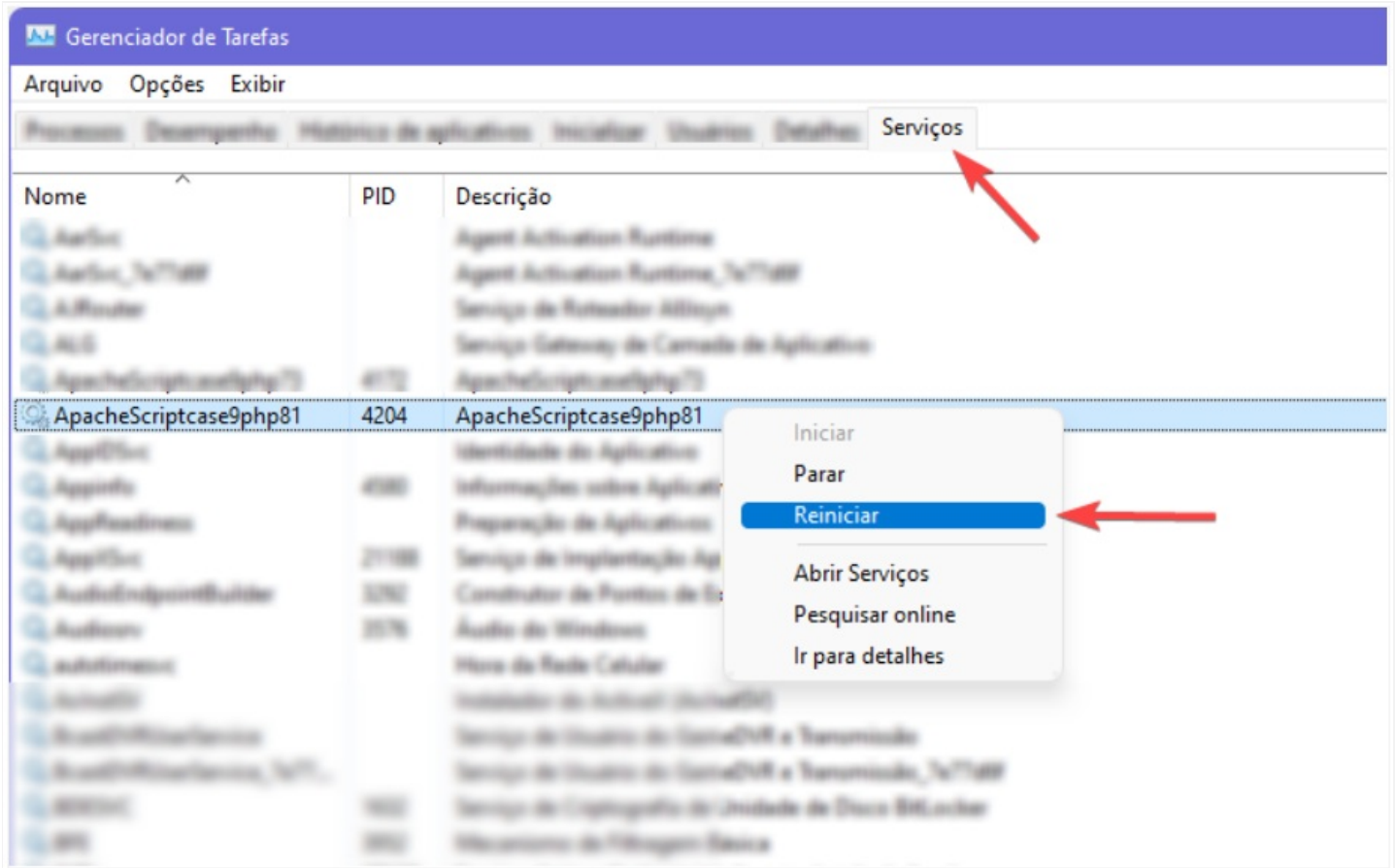
NOTE: If you are using an automatic Scriptcase installation, the SQLite driver will be enabled automatically.

1 - In the `php.ini` file, located under `C:\php`, uncomment the line for the SQLite extension **pdo_sqlite** by removing the `;` from the start of the line. See the example below:

```
922  ;extension=pdo_firebird
923  ;extension=pdo_mysql
924  ;extension=pdo_oci
925  ;extension=pdo_odbc
926  ;extension=pdo_pgsql
927  extension=pdo_sqlite
928  ;extension=pgsql
929  ;extension=shmop
```

2 - Restart the Apache service using the **Task Manager**.

- Open Task Manager and click on the **Services** tab.
- Look for the `Apache2.4` or `ApacheScriptcase9php73` service and right-click on this service, then **Restart**.

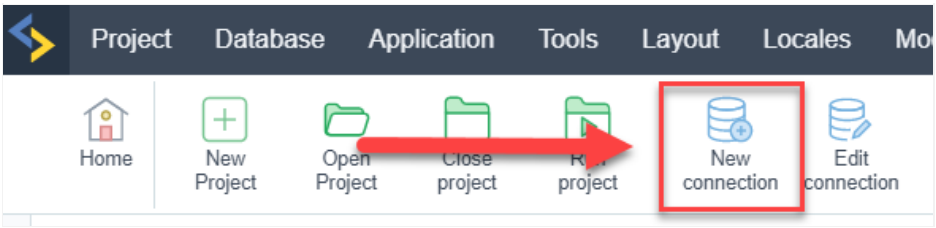


Creating a connection in Scriptcase

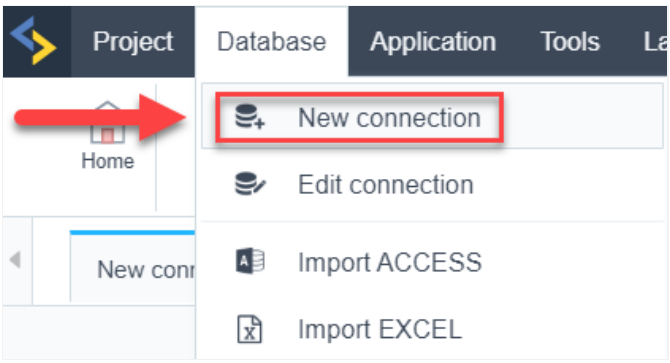
Here is how to create a connection in your Scriptcase project using the SQLite database.

1 - Access any project from within your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

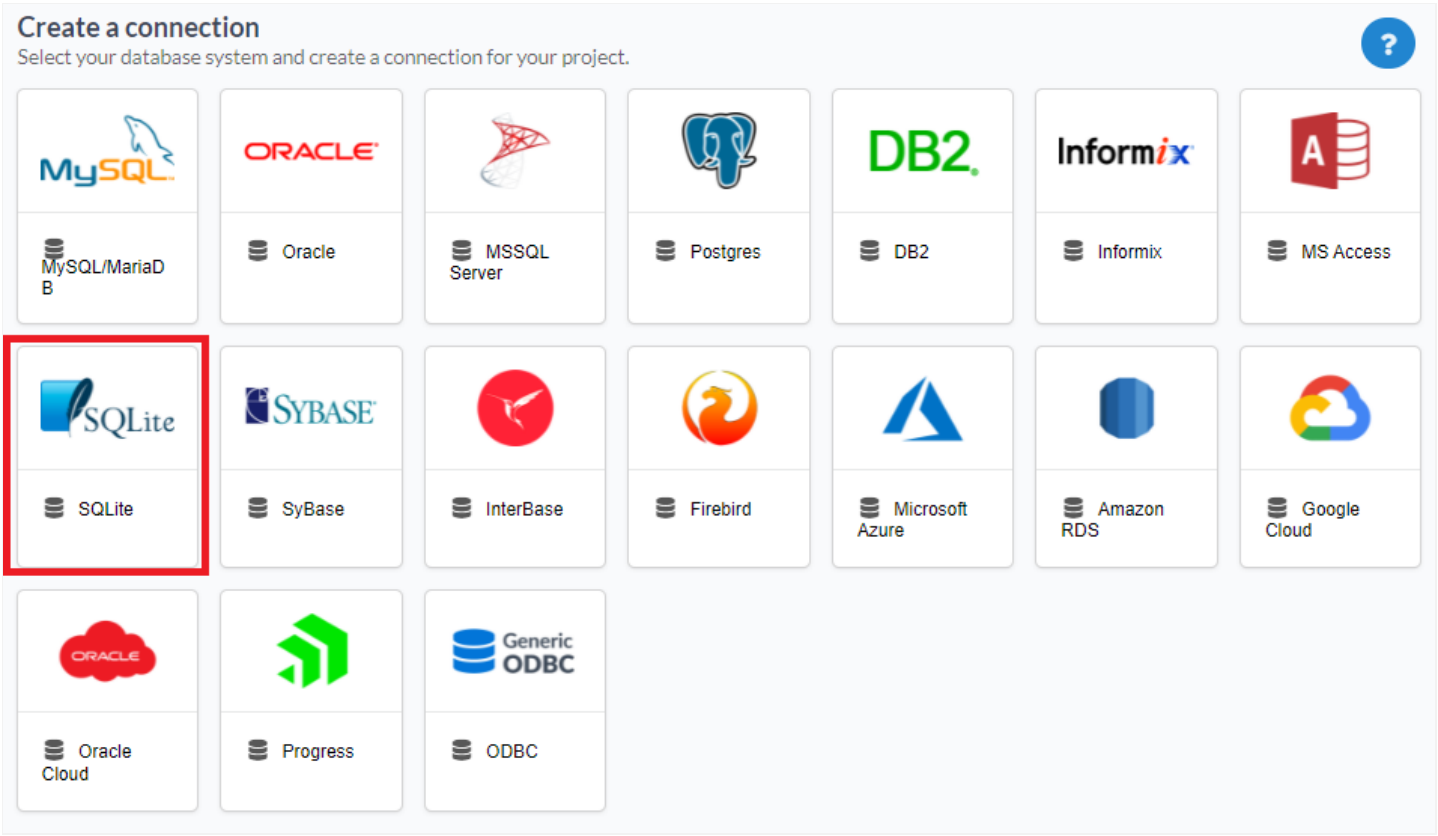


or go to the **Database > New Connection** menu.



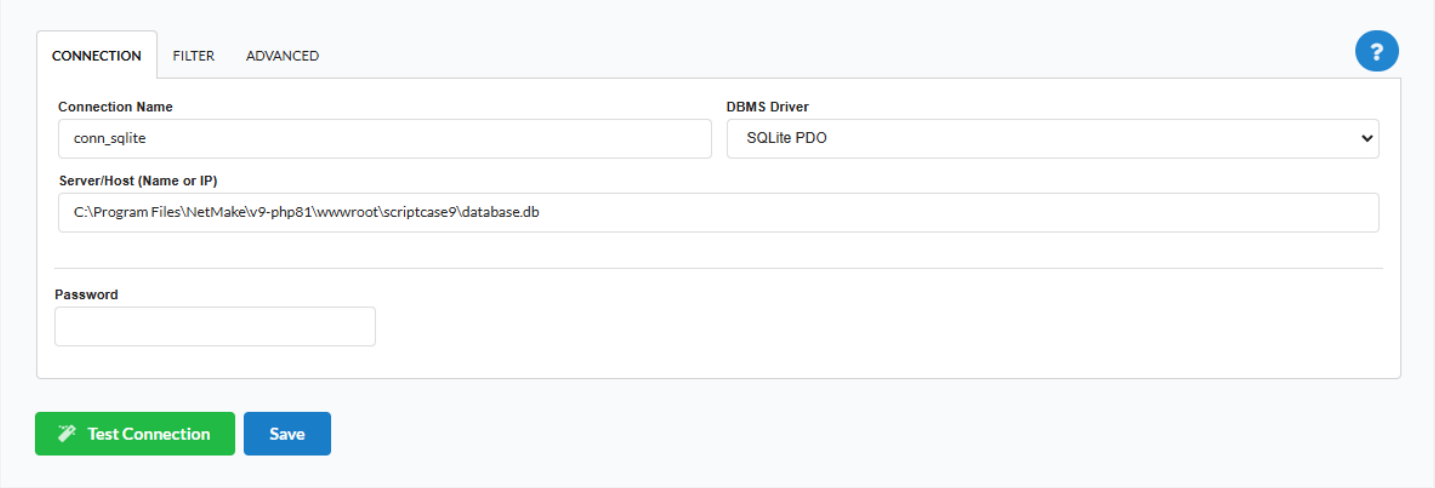
After that, a new page will appear with all the database connections.

3 - Select the **SQLite** connection.

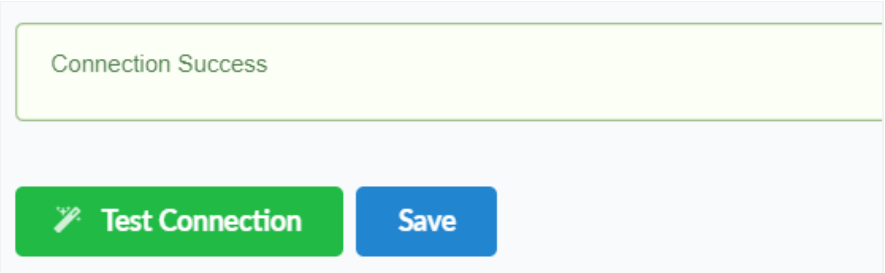


Connection

Enter the parameters for connecting to your SQLite database as follows:

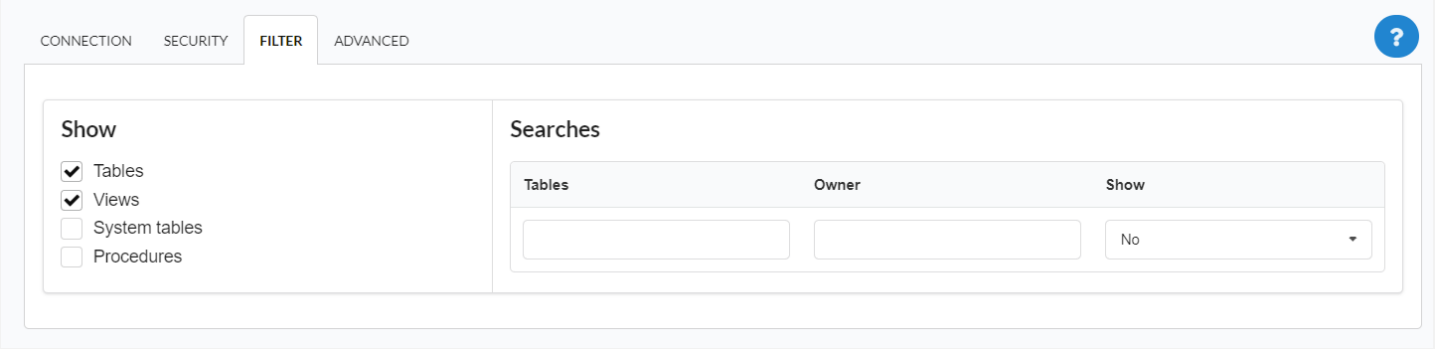


- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the SQLite Driver used to connect. By default, the driver will be **SQLite PDO**.
- **Server/Host(Name or IP)**: Enter the full path to your SQLite database file. **EX:** `C:\Apache24\htdocs\client_data`
- **Password**: Enter the password to complete the authentication process if you have it set for your DB file.
- **Test Connection**: Click this button to get a response from the Scriptcase request to see if the parameters you entered are correct.



Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.



Show

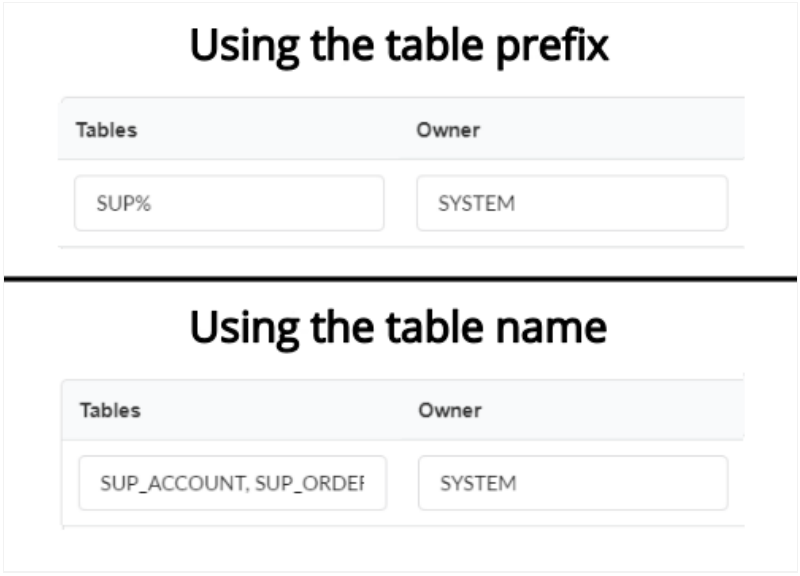
Allows the SQLite connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The setting can contain a `PREFIX%` or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

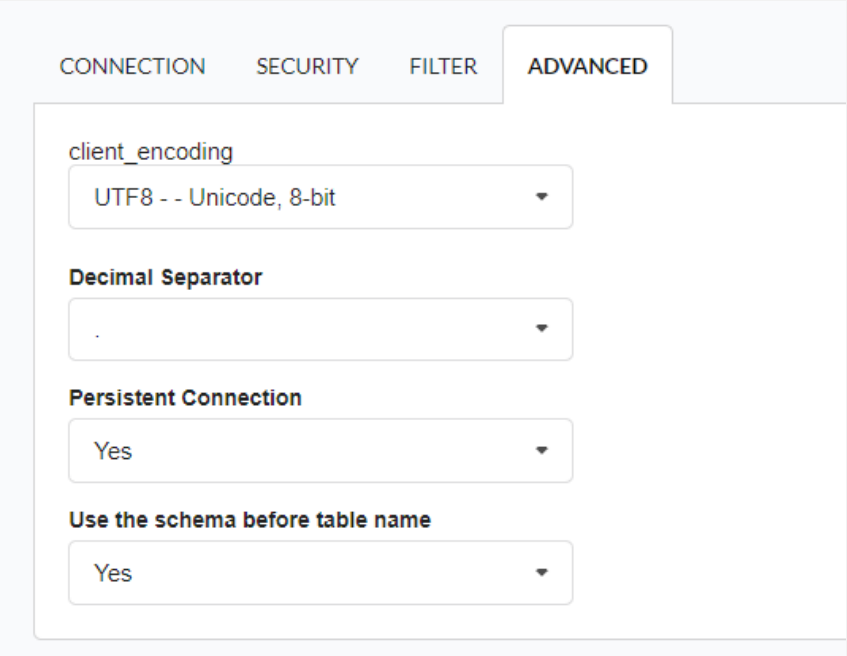


- **Owner:** Inform the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above**
- **Show:** Choose whether tables are displayed for the owner entered.

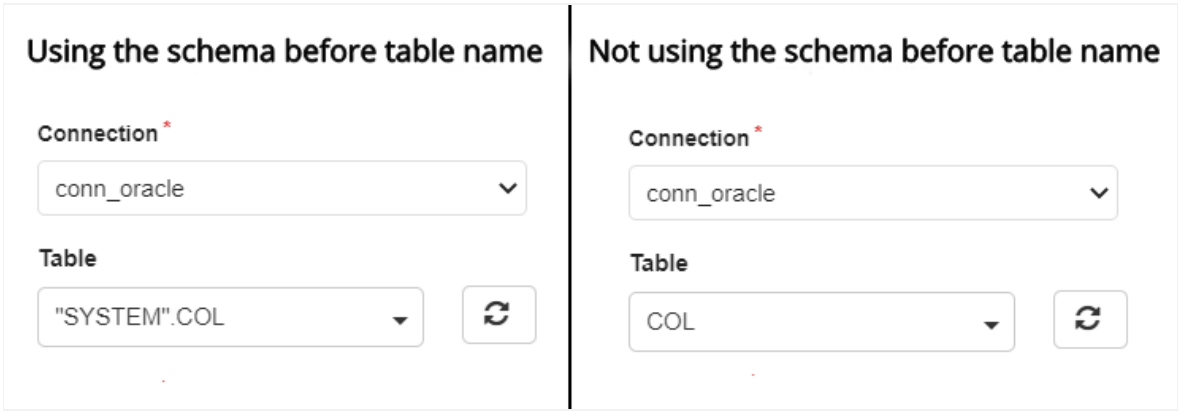
NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to settings specific to the connection. Changes made in this section impact the data display and performance of the application.

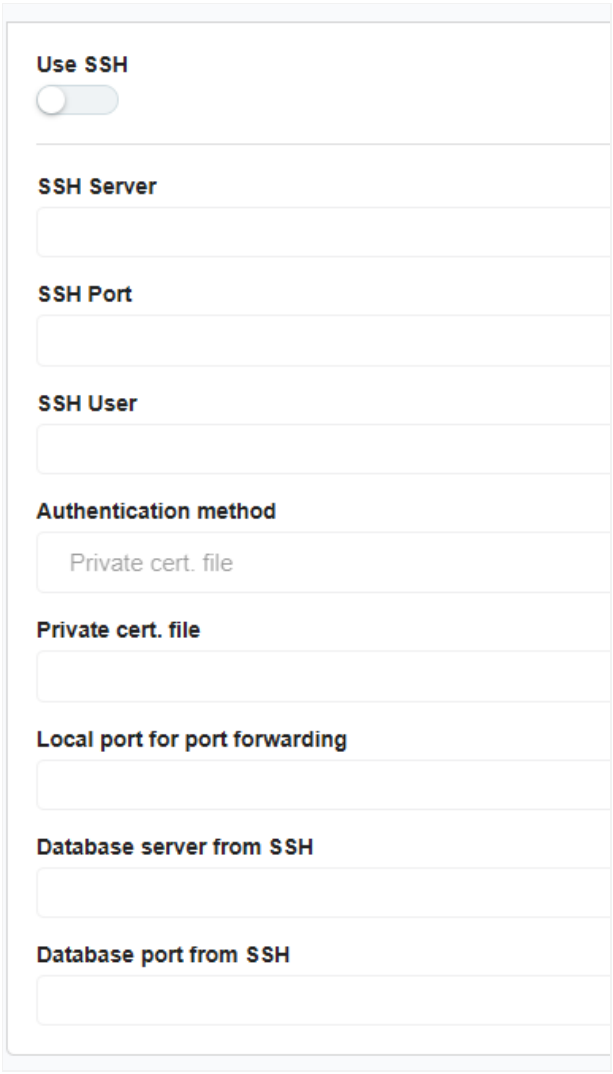


- **client_encoding:** Select the encoding used in your database. In the example above, we used client_encoding **UTF-8**.
- **Decimal separator:** Select the separator type for decimal records, between comma and period.
  - **By default, period `.` is selected as the separator.**
- **Persistent connection:** Set whether connections will be closed after your scripts run in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Set whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **Ex:**



SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

Linux

In Scriptcase, we only have one driver to connect to SQLite: **SQLite PDO**. If you are using your own pre-configured environment, the **SQLite extension must be enabled manually in PHP**.

Prerequisites

If you are using a manual installation on Linux, you will need to install the SQLite php extension. See below for how to do this:

NOTE: If you are using an automatic Scriptcase installation, the SQLite driver will be enabled automatically.

1 - Log into your **Linux** terminal and type this line below according to your operating system to install the SQLite PDO driver.

Ubuntu	CentOS
`sudo apt-get install php8.1-sqlite`	`sudo yum install php-sqlite`

2 - Make sure that the SQLite driver is enabled in your Scriptcase diagnostic:

- When you go to the top menu **Help > Diagnostics**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

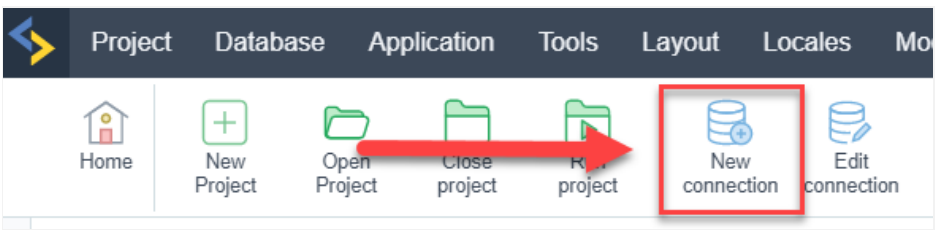
`127.0.0.1/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

Creating a connection in Scriptcase

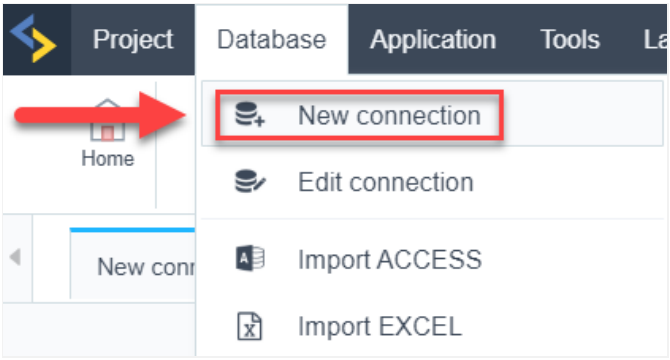
Here is how to create a connection in your Scriptcase project using the SQLite database.

1 - Access any project from within your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

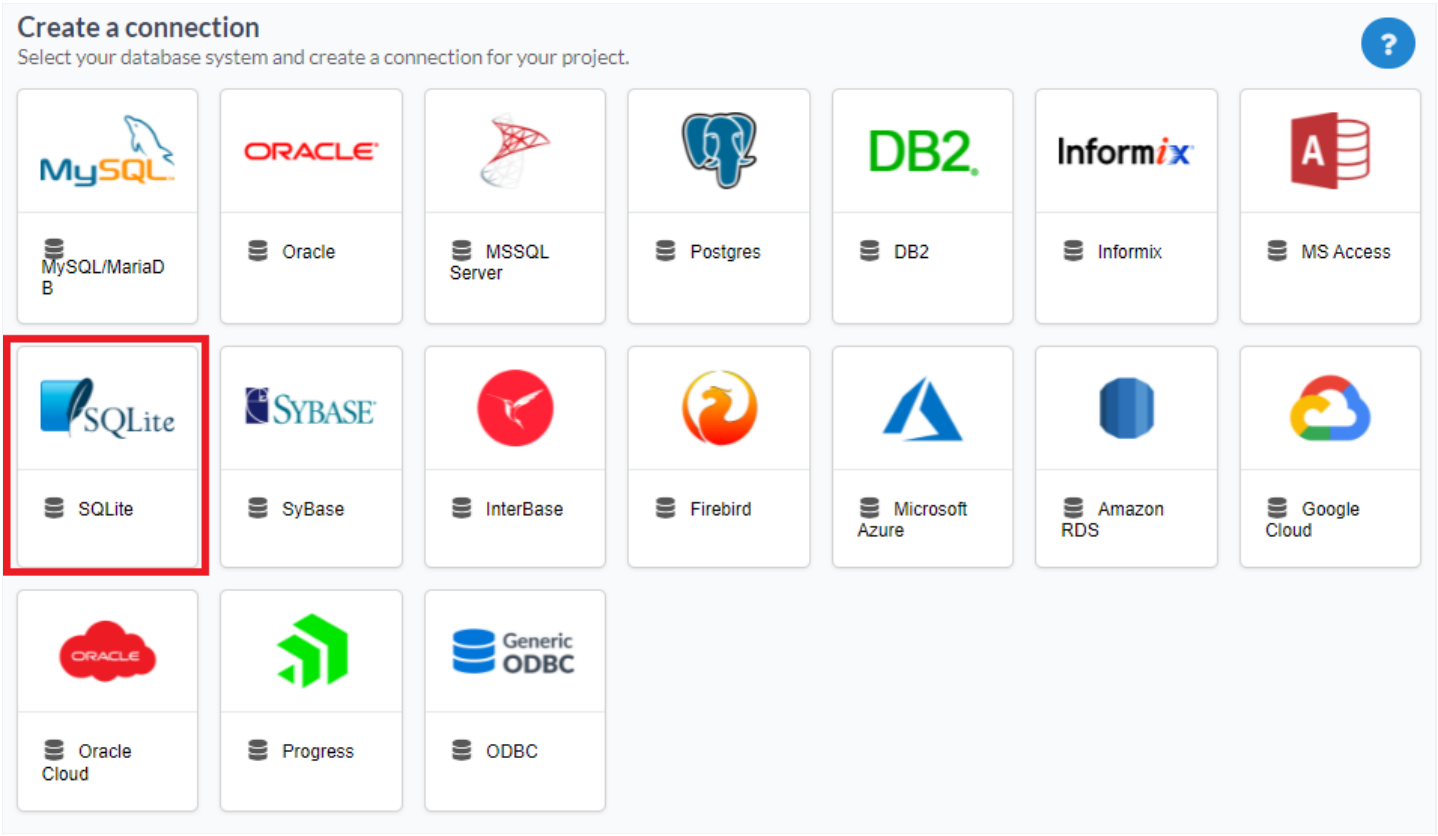


or go to the **Database > New Connection** menu.



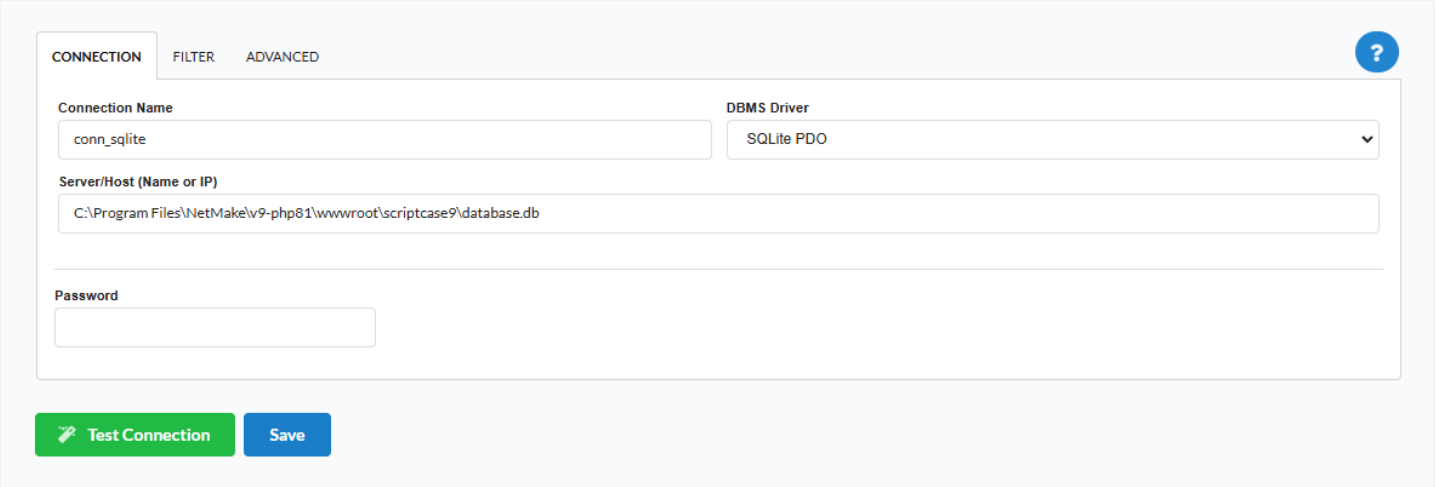
After that, a new page will appear with all the database connections.

3 - Select the **SQLite** connection.

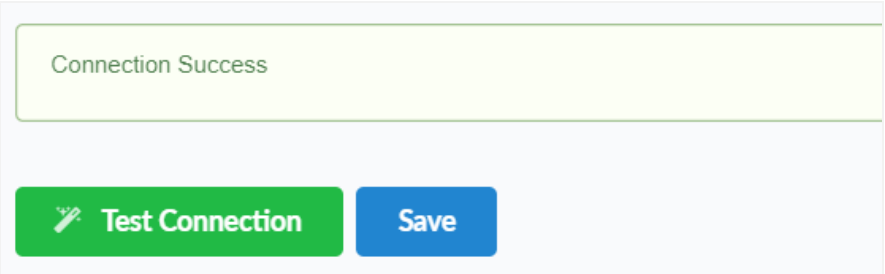


Connection

Enter the parameters for connecting to your SQLite database as follows:

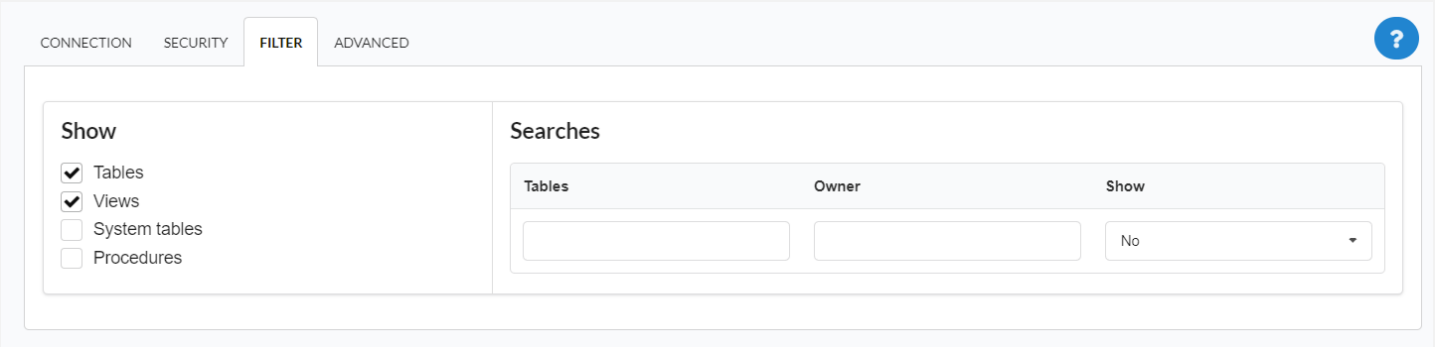


- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the SQLite Driver used to connect. By default, the driver will be **SQLite PDO**.
- **Server/Host(Name or IP)**: Enter the full path to your SQLite database file. **EX:** `C:\Apache24\htdocs\client_data`
- **Password**: Enter the password to complete the authentication process if you have it set for your DB file.
- **Test Connection**: Click this button to get a response from the Scriptcase request to see if the parameters you entered are correct.



Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.



Show

Allows the SQLite connection to see tables, views, system tables and procedures depending on the items selected by the user.

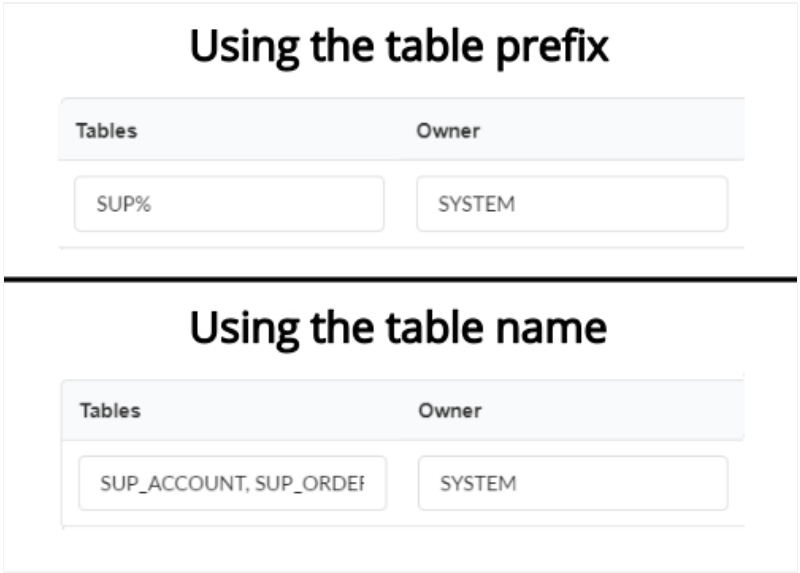
By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The setting can contain a `PREFIX%` or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

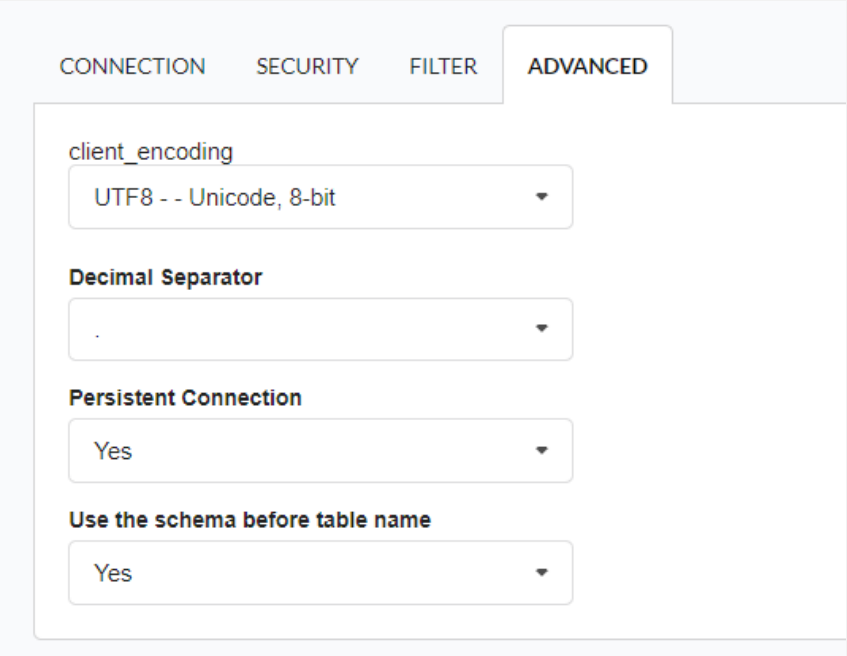


- **Owner:** Inform the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above**
- **Show:** Choose whether tables are displayed for the owner entered.

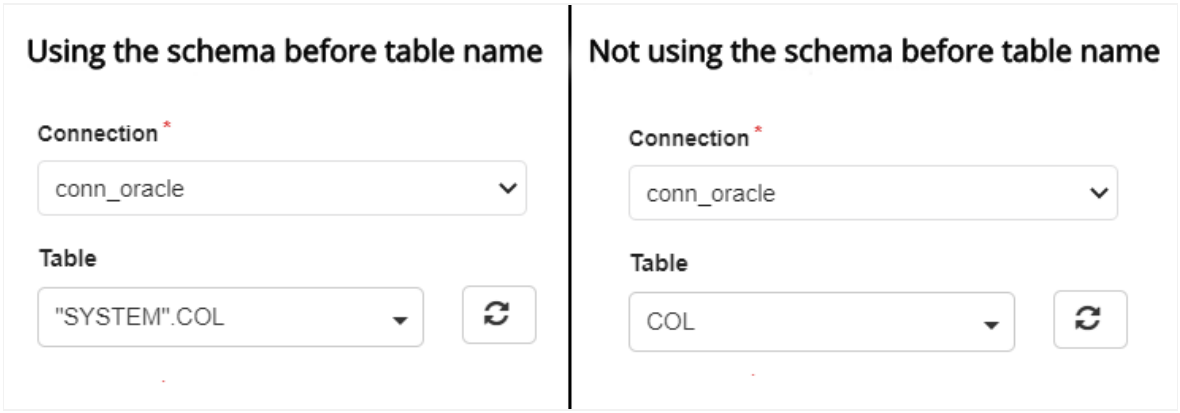
NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to settings specific to the connection. Changes made in this section impact the data display and performance of the application.

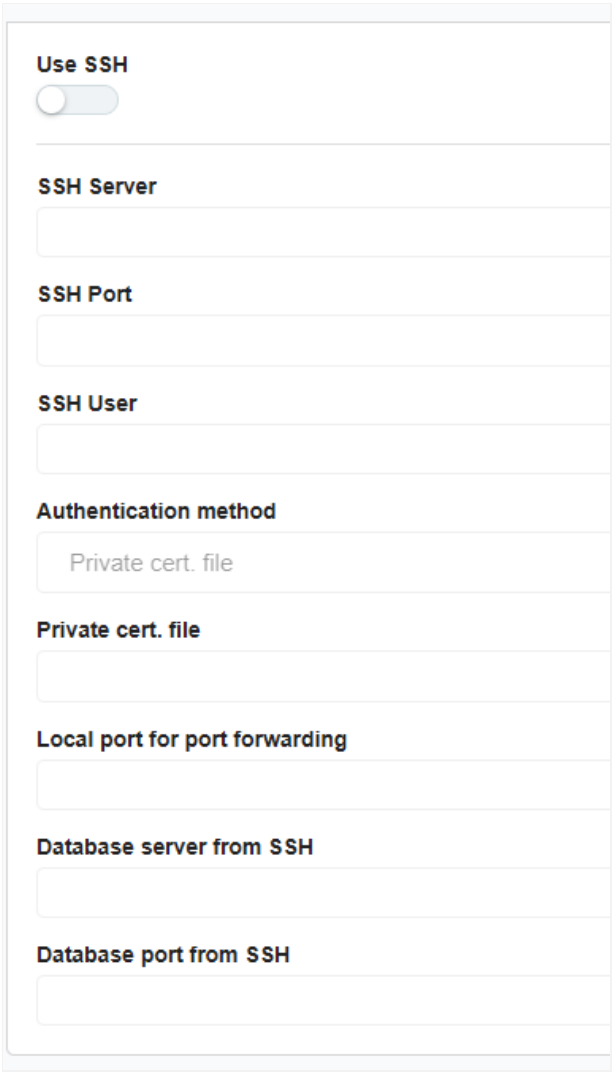


- **client_encoding:** Select the encoding used in your database. In the example above, we used client_encoding **UTF-8**.
- **Decimal separator:** Select the separator type for decimal records, between comma and period.
  - **By default, period `.` is selected as the separator.**
- **Persistent connection:** Set whether connections will be closed after your scripts run in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Set whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **Ex:**



SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

Mac OS

In Scriptcase, we only have one driver to connect to SQLite: **SQLite PDO**. If you are using your own pre-configured environment, the **SQLite extension must be enabled manually in PHP**.

Prerequisites

If you are using a manual installation on MacOS, you will need to install PHP and the SQLite driver will be enabled. Click here to see how to do this.

NOTE: If you are using an automatic Scriptcase installation, the SQLite driver will be enabled automatically.

1 - Verify that the SQLite driver is enabled in your Scriptcase diagnostic. See below for how to locate your diagnostic:

- When you go to the top menu **Help > Diagnostics**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

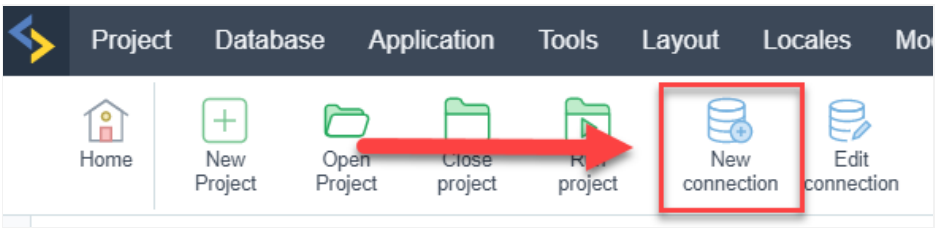
`127.0.0.1/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

Creating a connection in Scriptcase

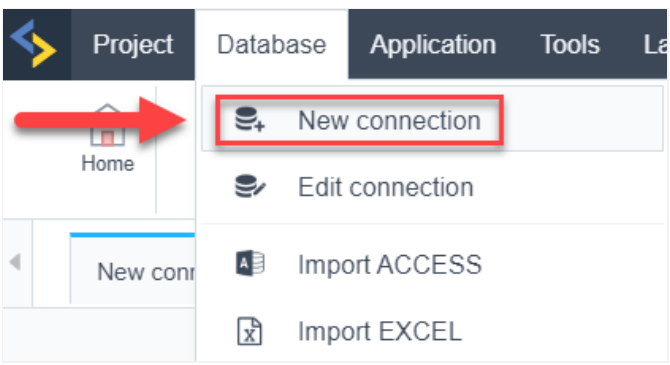
Here is how to create a connection in your Scriptcase project using the SQLite database.

1 - Access any project from within your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

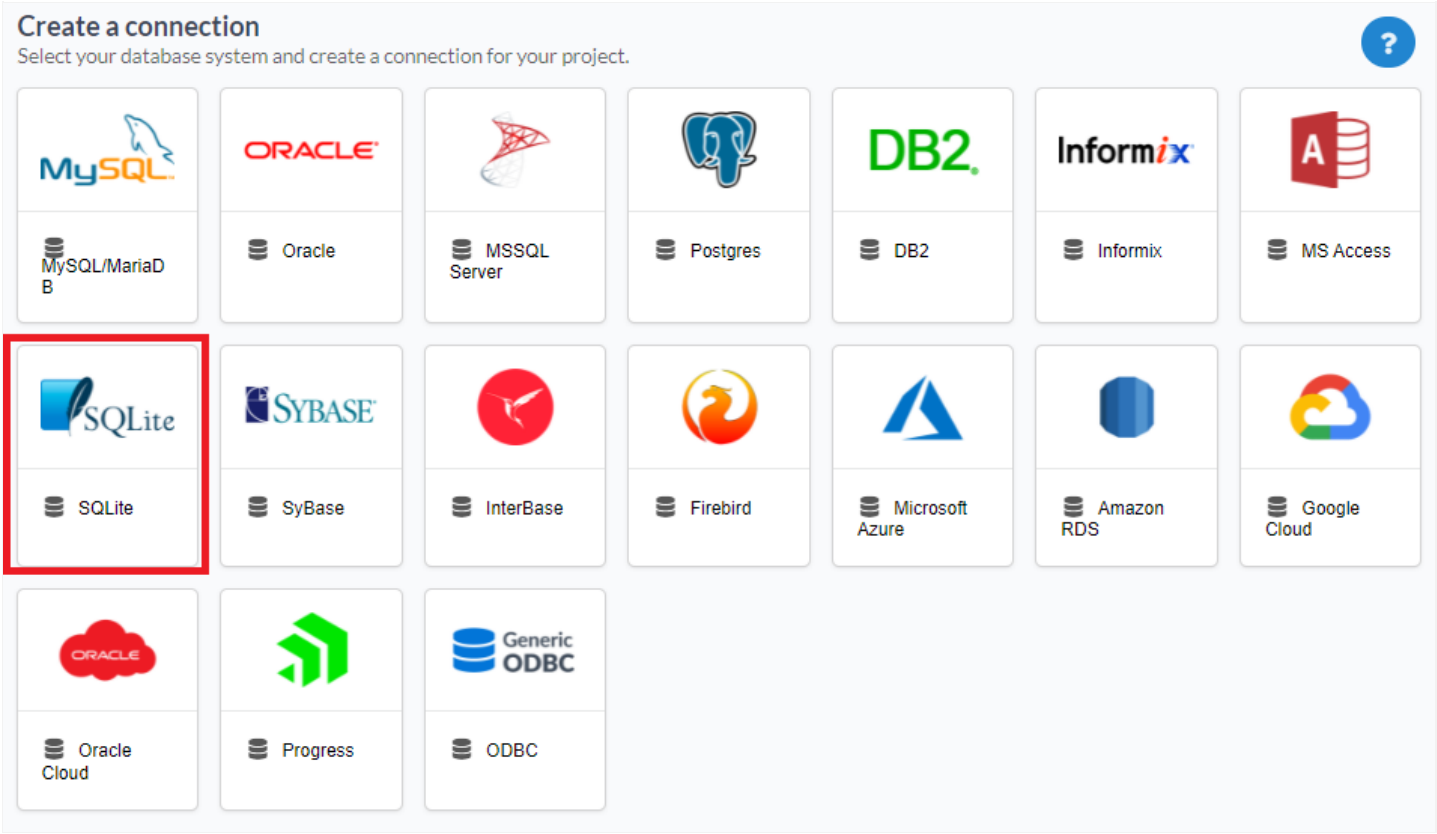


or go to the **Database > New Connection** menu.



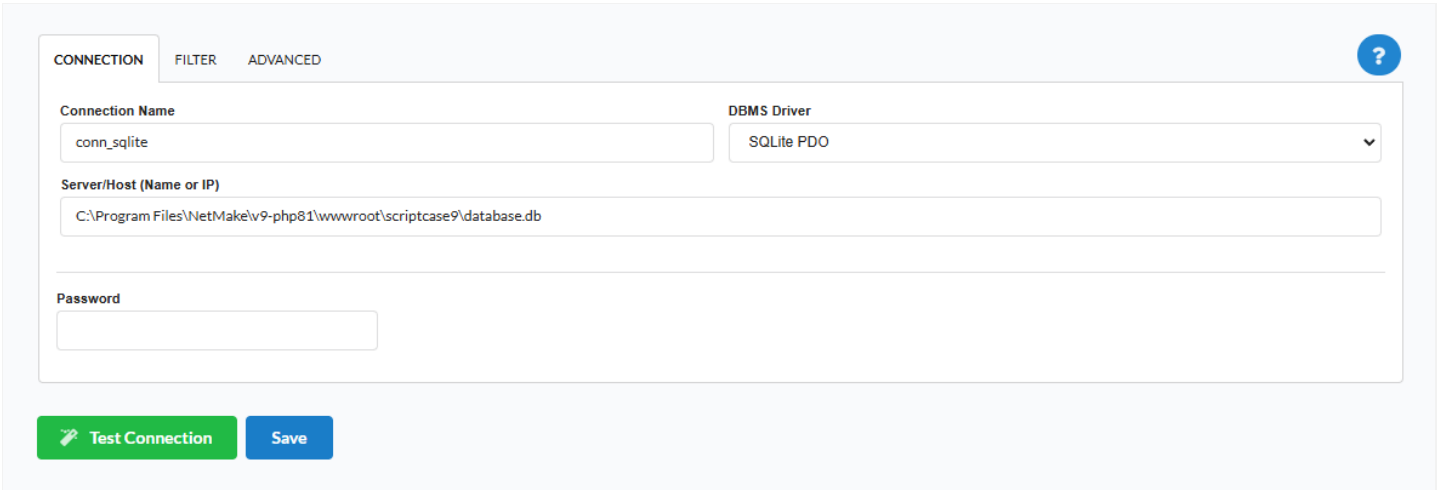
After that, a new page will appear with all the database connections.

3 - Select the **SQLite** connection.



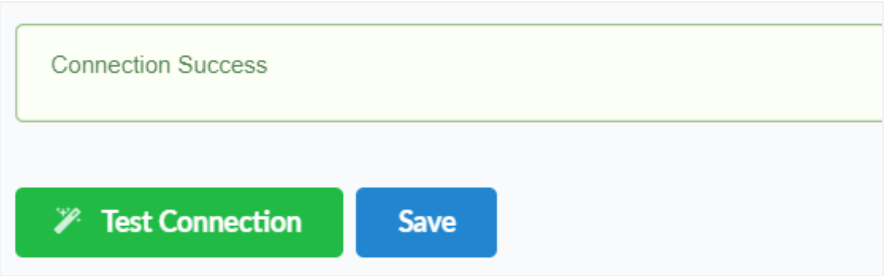
Connection

Enter the parameters for connecting to your SQLite database as follows:



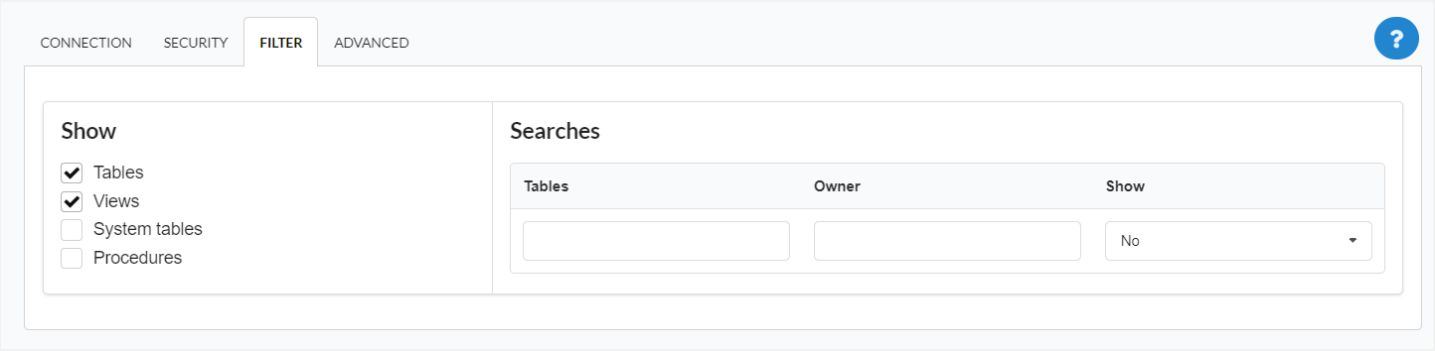
- **Connection Name**: Set the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the SQLite Driver used to connect. By default, the driver will be **SQLite PDO**.
- **Server/Host(Name or IP)**: Enter the full path to your SQLite database file. **EX:** `C:\Apache24\htdocs\client_data`
- **Password**: Enter the password to complete the authentication process if you have it set for your DB file.

- **Test Connection**: Click this button to get a response from the Scriptcase request to see if the parameters you entered are correct.



Filter

By accessing this tab, you can configure which items from the Database will be displayed in the connection, depending on the owner or not.



Show

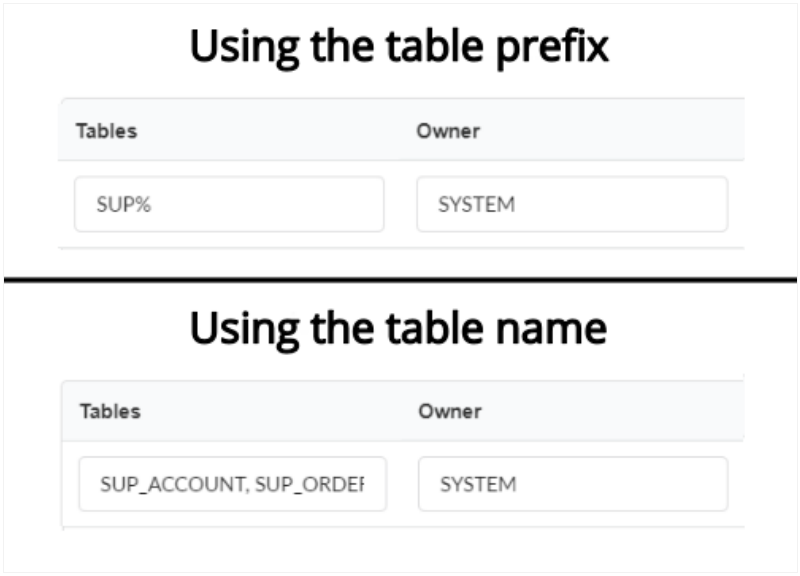
Allows the SQLite connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** When you select this option, the procedures from your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The setting can contain a `PREFIX%` or name of the tables to be displayed.
  - **By default, Scriptcase leaves this option empty.**
  - **Ex:**

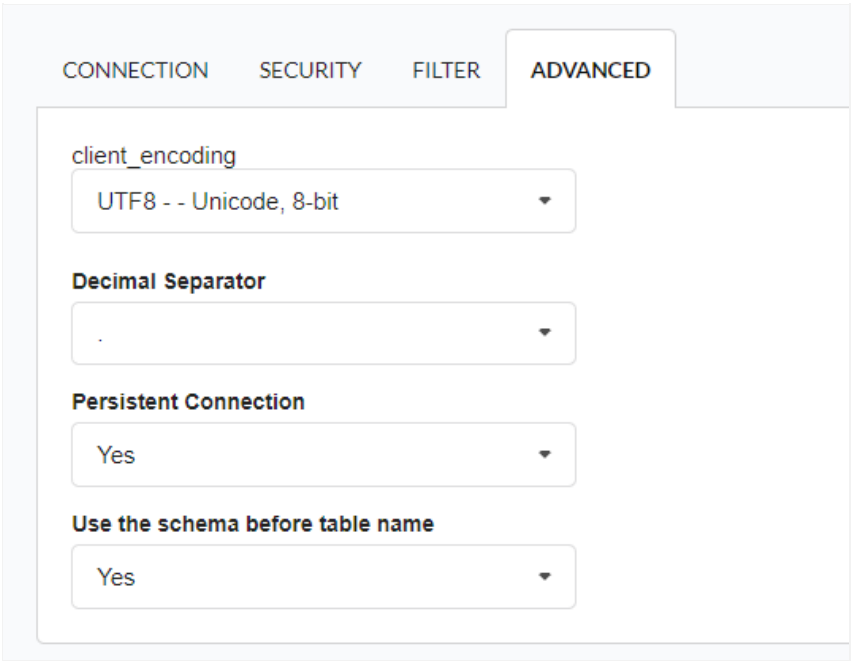


- **Owner:** Inform the user who sees the inserted tables for display.
  - **User must be in capital letters, as in the example above**
- **Show:** Choose whether tables are displayed for the owner entered.

NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

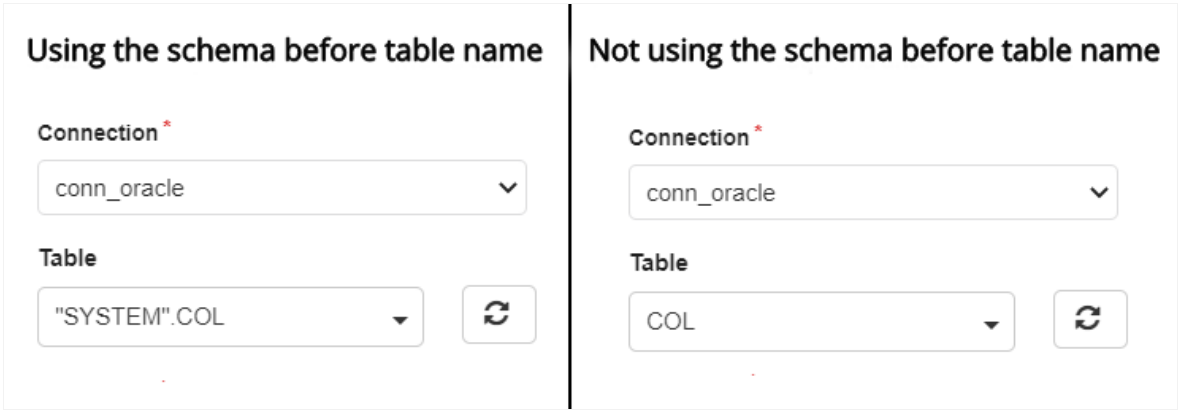
In this tab, you have access to settings specific to the connection. Changes made in this section impact the data display and performance of the application.



- **client_encoding:** Select the encoding used in your database. In the example above, we used client_encoding **UTF-8**.
- **Decimal separator:** Select the separator type for decimal records, between comma and period.
  - **By default, period `.` is selected as the separator.**

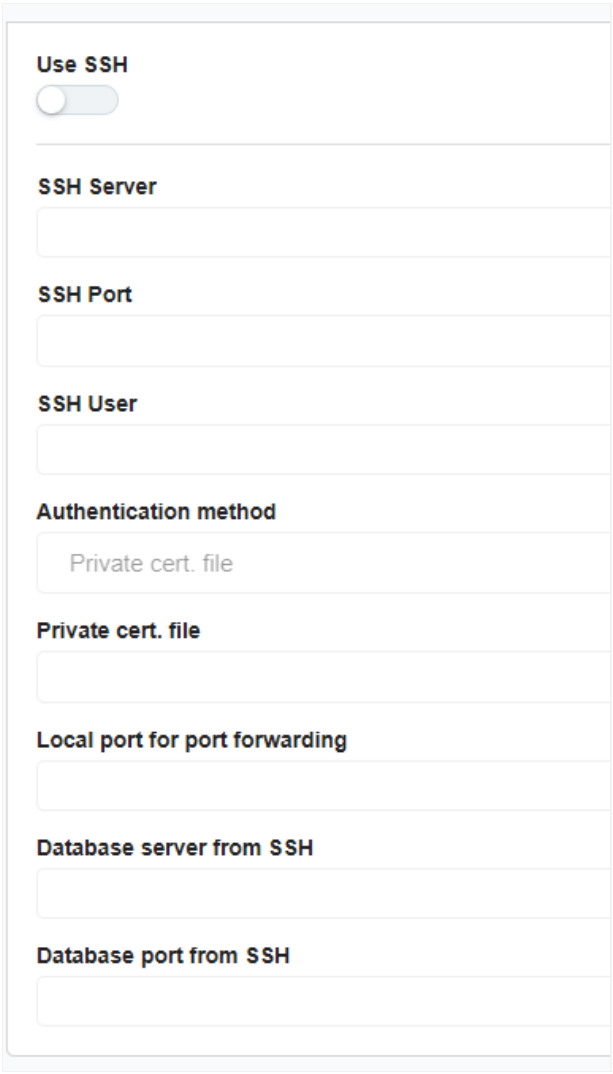
- **Persistent connection:** Set whether connections will be closed after your scripts run in Scriptcase applications.
  - **By default, Scriptcase disables this option.**

- **Use the schema before table name:** Set whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **Ex:**



SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

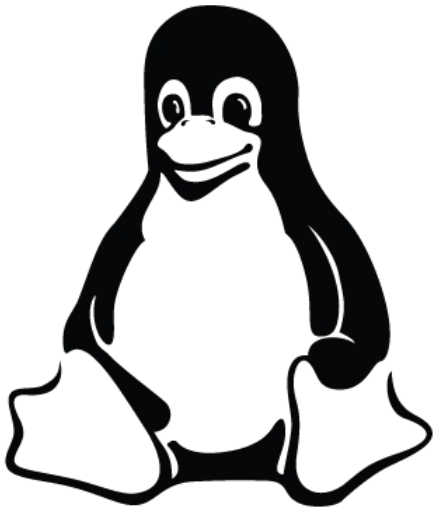
Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

ODBC



Windows

The ODBC Generic connection is normally used to create connections with the databases: dBase (.dbf), Dataflex, Paradox, Visual FoxPro, etc.

You only need have the corresponding ODBC driver with the database installed.

If you are using your own previously configured environment, you will need to enable the ODBC extension in PHP.

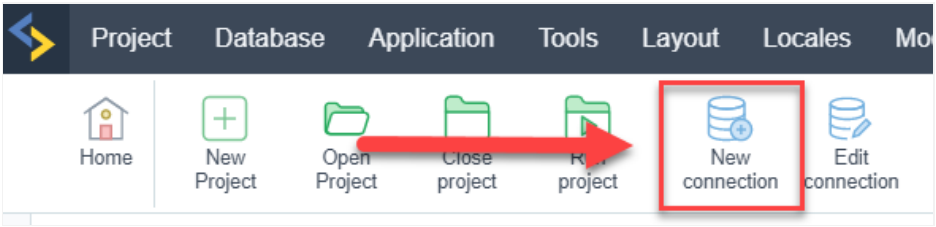
Generic ODBC

Setting up ODBC on Windows

The automatic installation already comes with the ODBC only needing to connect with the database with ScriptCase.

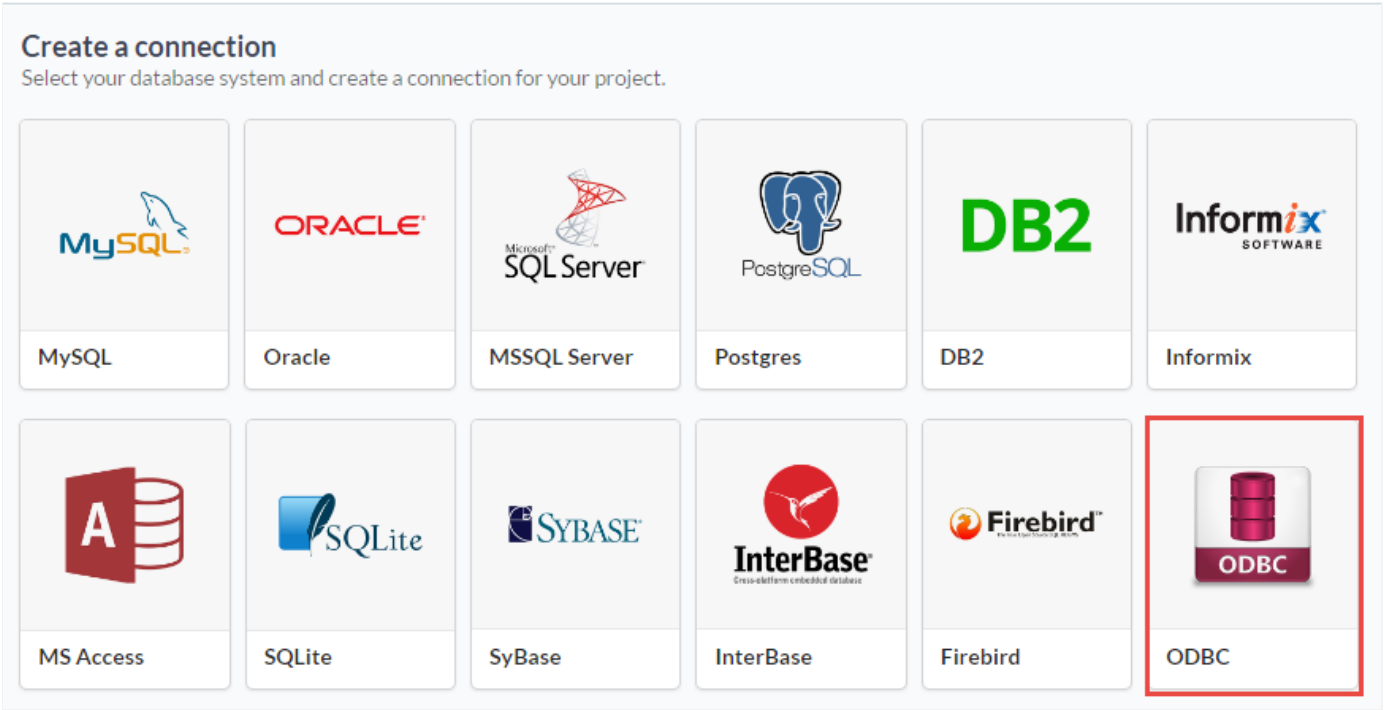
Creating a Connection with ScriptCase

- Access ScriptCase using the URL http://127.0.0.1:8092/scriptcase
- After accessing ScriptCase, access or create a new project, click on the new connection icon or access “Database > New Connection”



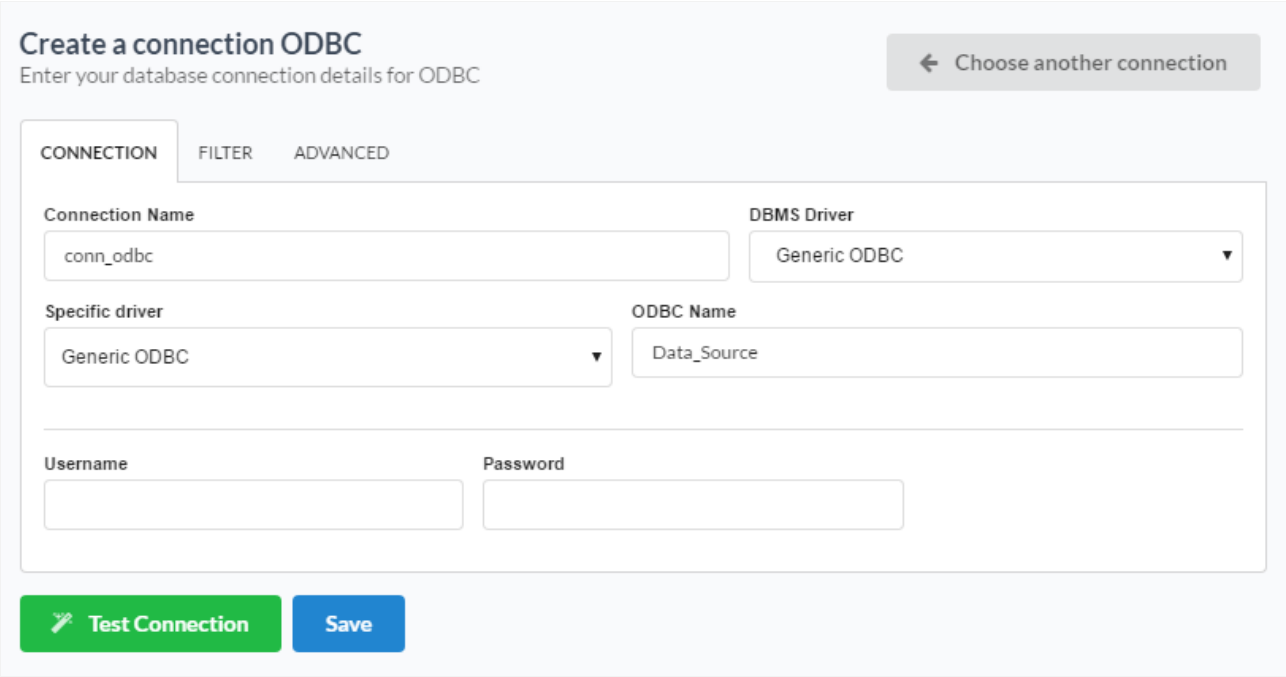
Creating a new connection

- After that, you will see a page with all the database connections that you can create. Select a “ODBC” connection



Selecting a Windows database connection

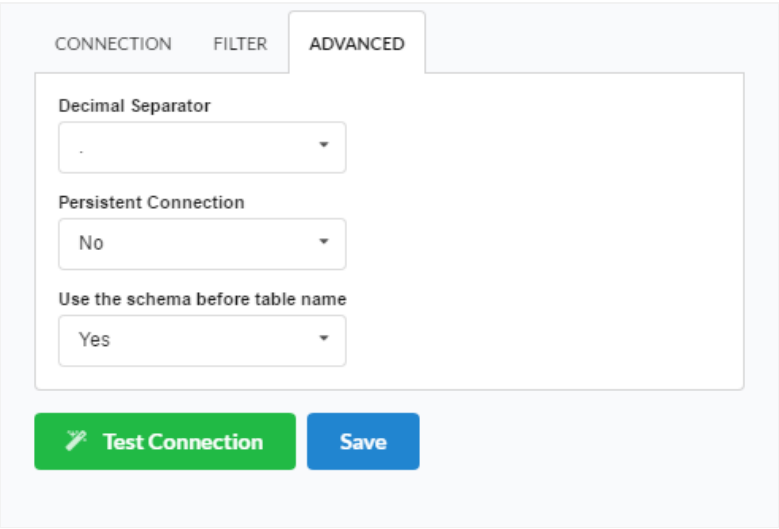
- **New connection** : You will place the information of the connection to your database here:



Connecting with Windows database

- **Connection Name** : Defines the name of your new connection.
- **DBMS Driver** : Select the Generic ODBC.
- **Specific driver** : Defined as Generic ODBC, allows you to connect to specific Data Sources: DBF, FileMaker, Progress or Visual FoxPro.
- **ODBC Name** : On this option, you will inform the name of the Data Source that you have created.
- **Username** : Inform the username of the Windows database, only if there are any.
- **Password** : Inform the password of the Windows database, only if there are any.
- **Test Connection** : Displays a message of the status of the connection if successful or not.

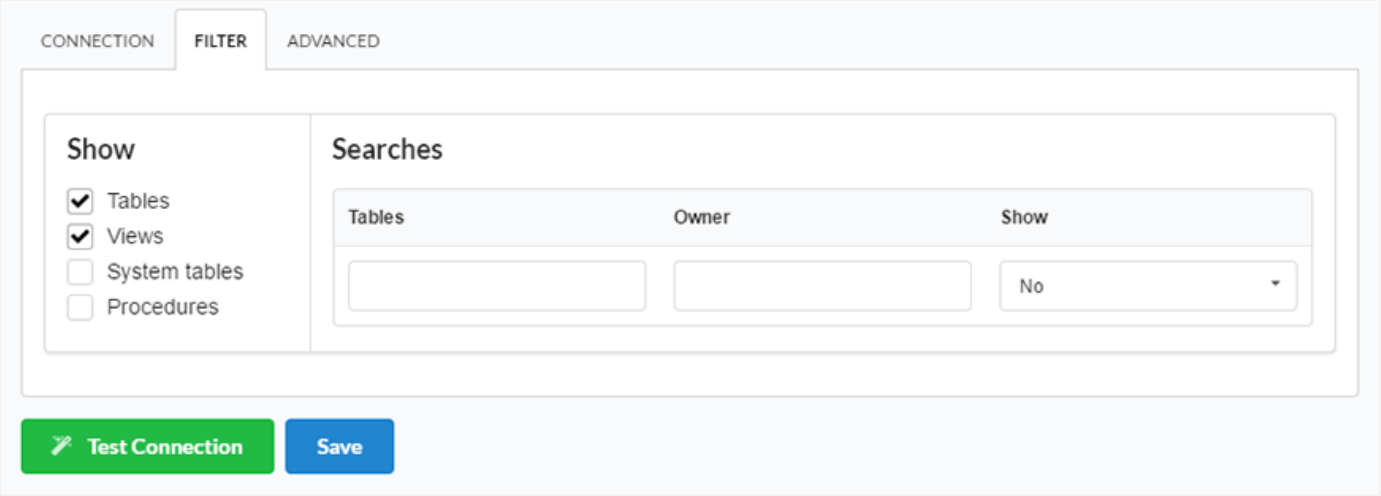
- For more options on connecting, click on the **Advanced** tab:



Advanced setup for the Windows database

- **Decimal Separator** - Select the separator type, between a dot or a comma.
- **Persistent Connection** - Persistent Connection are connections that do not close when finishing a script.
- **Use the schema before table name** - Allows the use of schema before the name of the tables.

- **Filter** : Accessing this tab, we can setup which tables will be listed on this connection:



Filtering the Windows database

- **Show** - Allows the display of the the filters on the tables, views, System Tables and procedures.
  - **Tables** - Allows to setup the display of Tables from your database.
  - **Views** - Allows to setup the display of Views from your database.
  - **System Tables** - Allows to setup the display of System Tables from your database.
  - **Procedures** - Allows to setup the display of Procedures from your database.
- **Searches** - Allows to define which tables and Owner will be displayed.
  - **Tables** - Allows to define a prefix (prefix%) or name of the tables for display.
  - **Owner** - Allows to define the owner of the tables for listing.
  - **Show** - Allows to define what will be displayed or not from the table an owner's setup.

Advanced

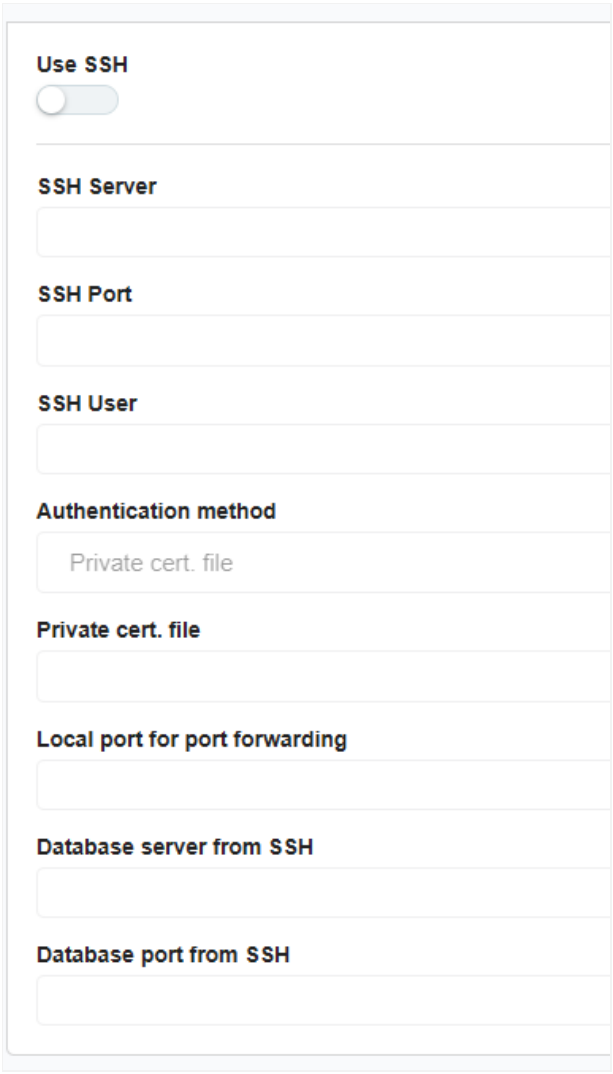
In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

![Advanced database connection configuration][conexao_avancado]

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the `.` dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

.png

Linux

The ODBC Generic connection is normally used to create connections with the databases: dBase (.dbf), Dataflex, Paradox, Visual FoxPro, etc.

You only need have the corresponding ODBC driver with the database installed.

If you are using your own previously configured environment, you will need to enable the ODBC extension in PHP.

Generic ODBC

Setting up ODBC on Linux

The automatic installation already comes with the ODBC only needing to install the UnixODBC driver so that you can do the connection.

UBUNTU\DEBIAN x86_64	CENTOS\RHEL x86_64
- sudo apt-get update	- sudo yum update
- sudo apt-get install unixodbc-dev unixodbc	- sudo yum install unixODBC

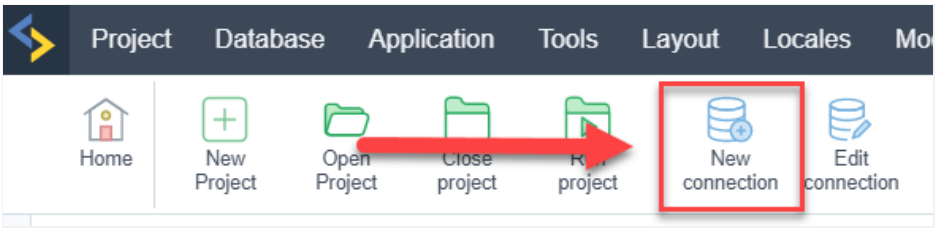
UBUNTU\DEBIAN i386 (x86)	CENTOS\RHEL i686 (x86)
- sudo apt-get update	- sudo yum update
- sudo apt-get install unixodbc-dev unixodbc	- sudo yum install unixODBC

Restart the Apache Server:

```
sudo /etc/init.d/apachesc9 restart
```

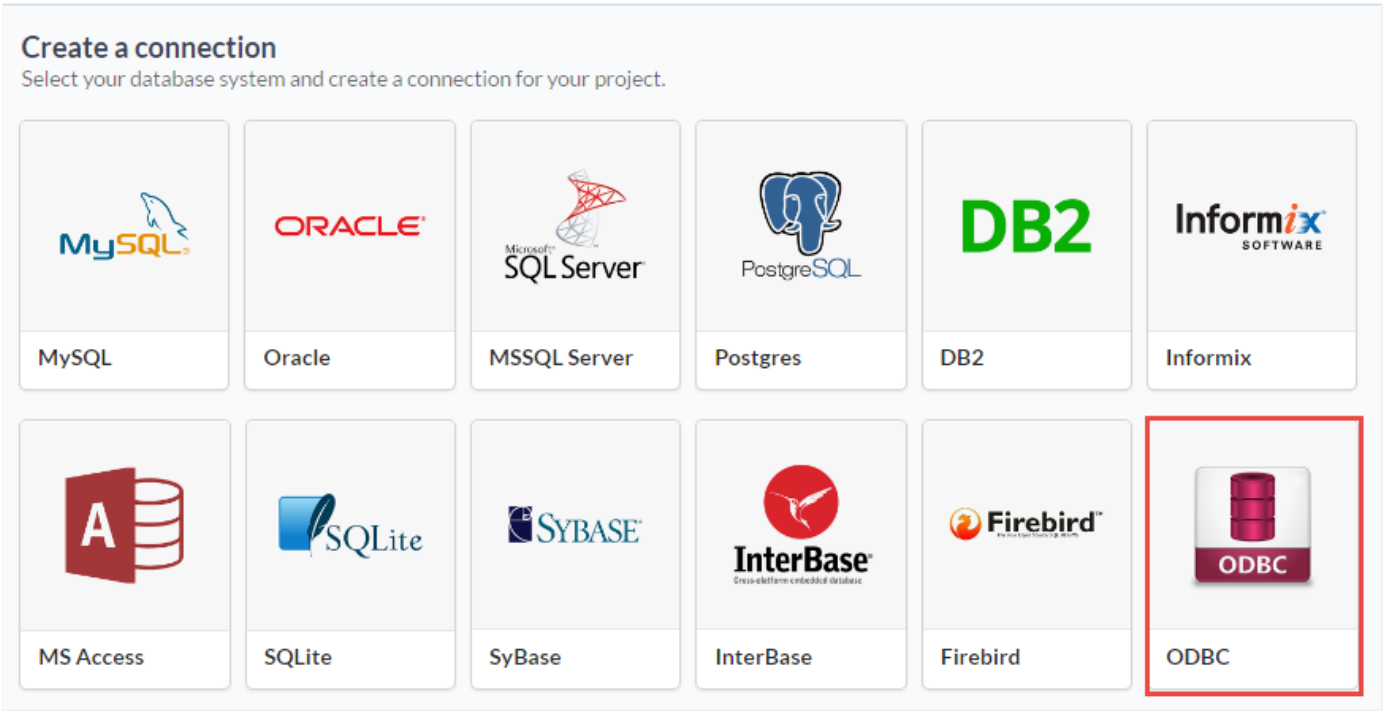
Creating a Connection with ScriptCase

- Access ScriptCase using the URL http://127.0.0.1:8092/scriptcase
- After accessing ScriptCase, access or create a new project, click on the new connection icon or access “Database > New Connection”



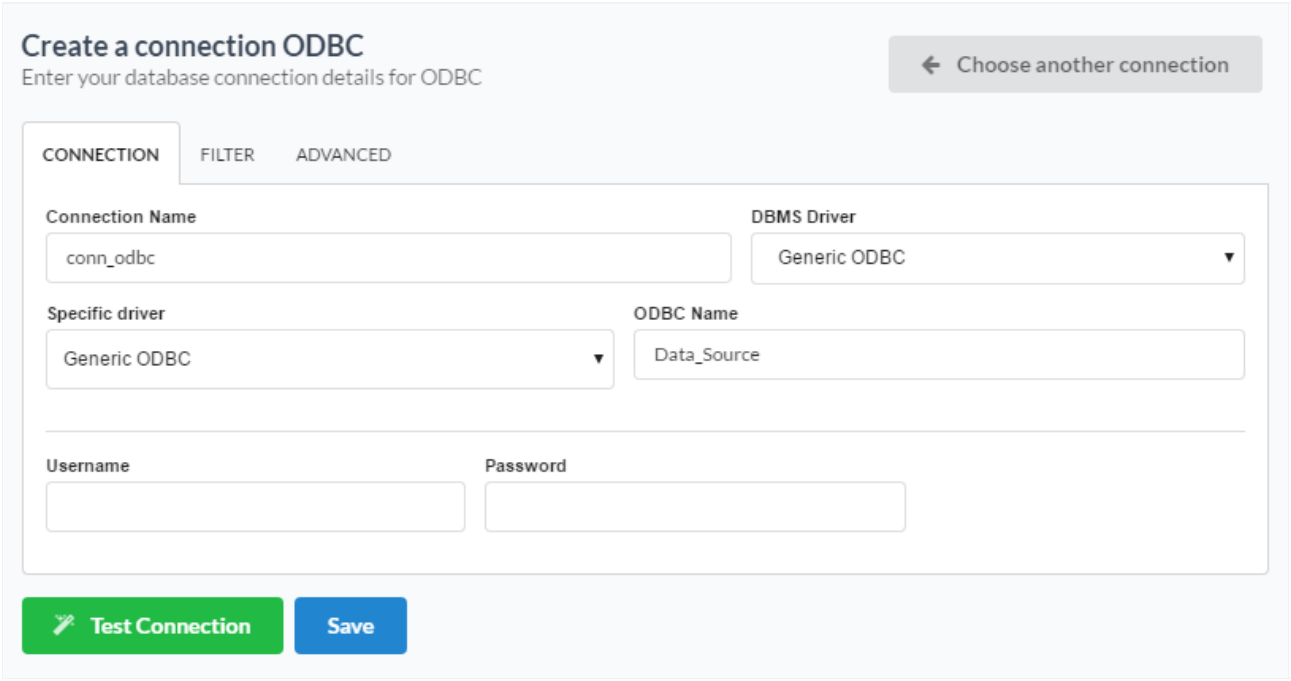
Creating a new connection

- After that, you will see a page with all the database connections that you can create. Select a “ODBC” connection



Selecting a Linux database connection

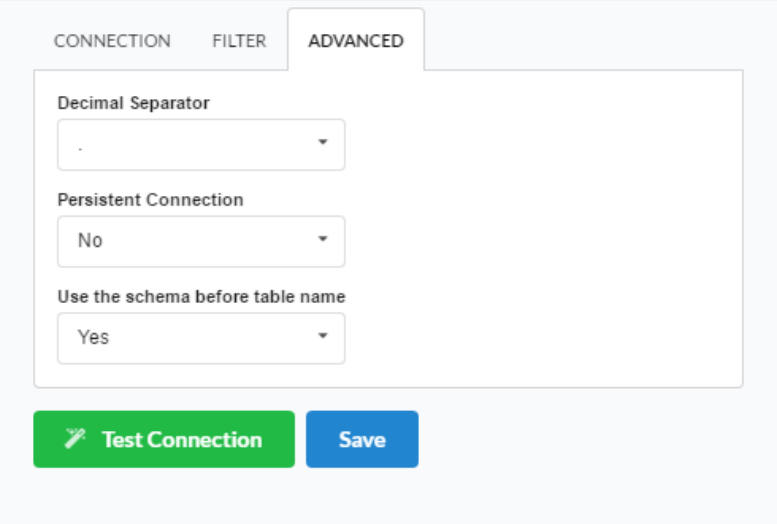
- **New connection** : You will place the information of the connection to your database here:



Connecting with Linux database

- **Connection Name** : Defines the name of your new connection.
- **DBMS Driver** : Select the Generic ODBC.
- **Specific driver** : Defined as Generic ODBC, allows you to connect to specific Data Sources: DBF, FileMaker, Progress or Visual FoxPro.
- **ODBC Name** : On this option, you will inform the name of the Data Source that you have created.
- **Username** : Inform the username of the Linux database, only if there are any.
- **Password** : Inform the password of the Linux database, only if there are any.
- **Test Connection** : Displays a message of the status of the connection if successful or not.

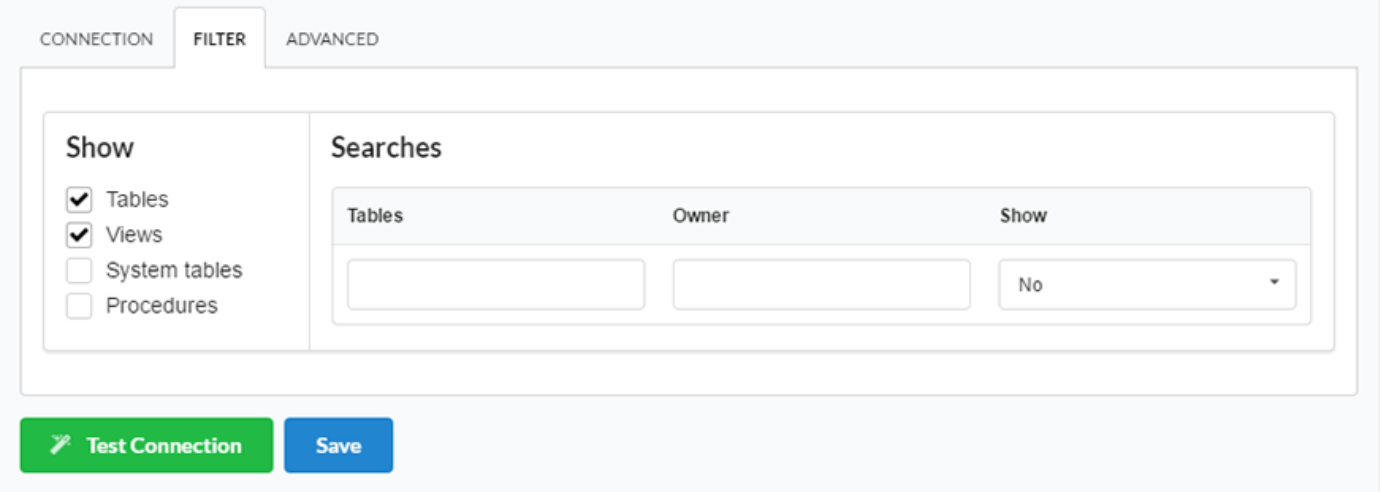
- For more options on connecting, click on the **Advanced** tab:



Advanced setup for the Linux database

- **Decimal Separator** - Select the separator type, between a dot or a comma.
- **Persistent Connection** - Persistent Connection are connections that do not close when finishing a script.
- **Use the schema before table name** - Allows the use of schema before the name of the tables.

- **Filter** : Accessing this tab, we can setup which tables will be listed on this connection:



Filtering the Linux database

- **Show** - Allows the display of the the filters on the tables, views, System Tables and procedures.
  - **Tables** - Allows to setup the display of Tables from your database.
  - **Views** - Allows to setup the display of Views from your database.
  - **System Tables** - Allows to setup the display of System Tables from your database.
  - **Procedures** - Allows to setup the display of Procedures from your database.
- **Searches** - Allows to define which tables and Owner will be displayed.
  - **Tables** - Allows to define a prefix (prefix%) or name of the tables for display.
  - **Owner** - Allows to define the owner of the tables for listing.
  - **Show** - Allows to define what will be displayed or not from the table an owner's setup.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

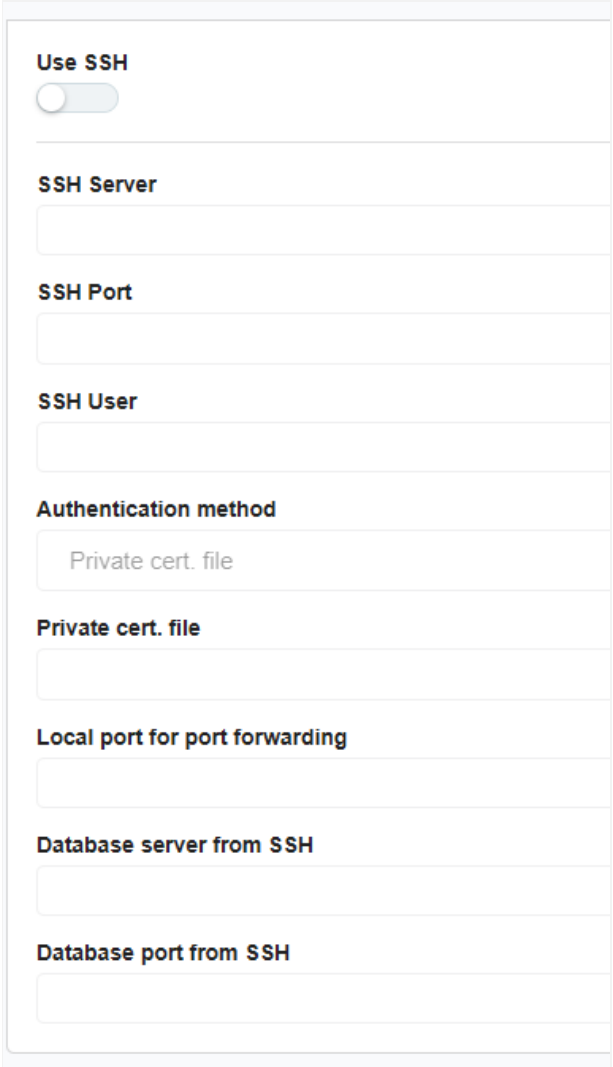
![Advanced database connection configuration][conexao_avancado]

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.

    - **By default, the `.` dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
    - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

.png

Mac OS X

The ODBC Generic connection is normally used to create connections with the databases: dBase (.dbf), Dataflex, Paradox, Visual FoxPro, etc.

You only need have the corresponding ODBC driver with the database installed.

If you are using your own previously configured environment, you will need to enable the ODBC extension in PHP.

Generic ODBC

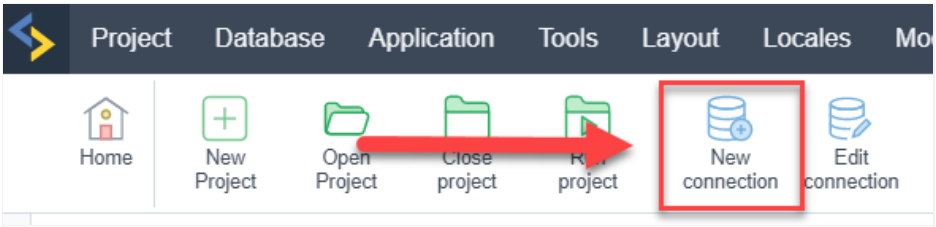
Settin gup ODBC on MAC

The automatic installation already comes with the ODBC driver, only needing to create the data source to connect with the database with ScriptCase.

The path of odbc.ini is: /Applications/Scriptcase/components/apache/bin/unixODBC/etc/

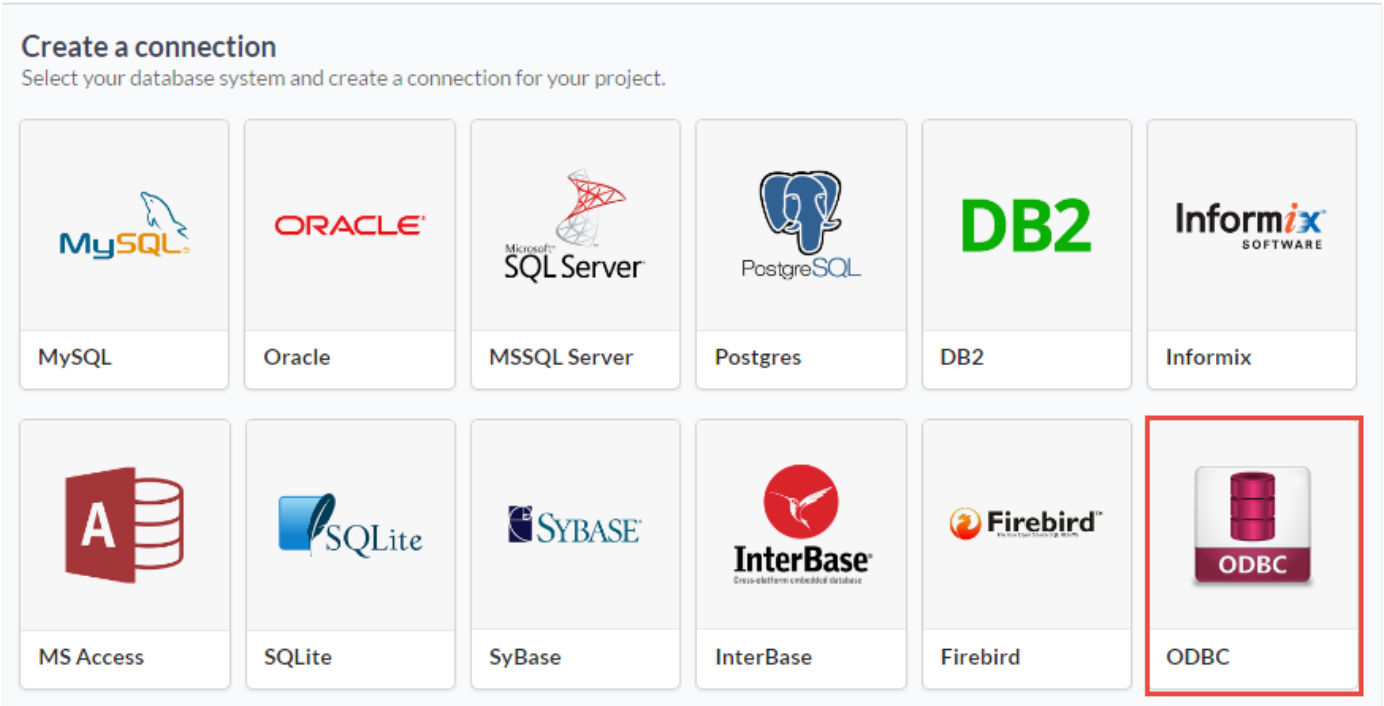
Creating a Connection with ScriptCase

- Access ScriptCase using the URL http://127.0.0.1:8092/scriptcase
- After accessing ScriptCase, access or create a new project, click on the new connection icon or access “Database > New Connection”



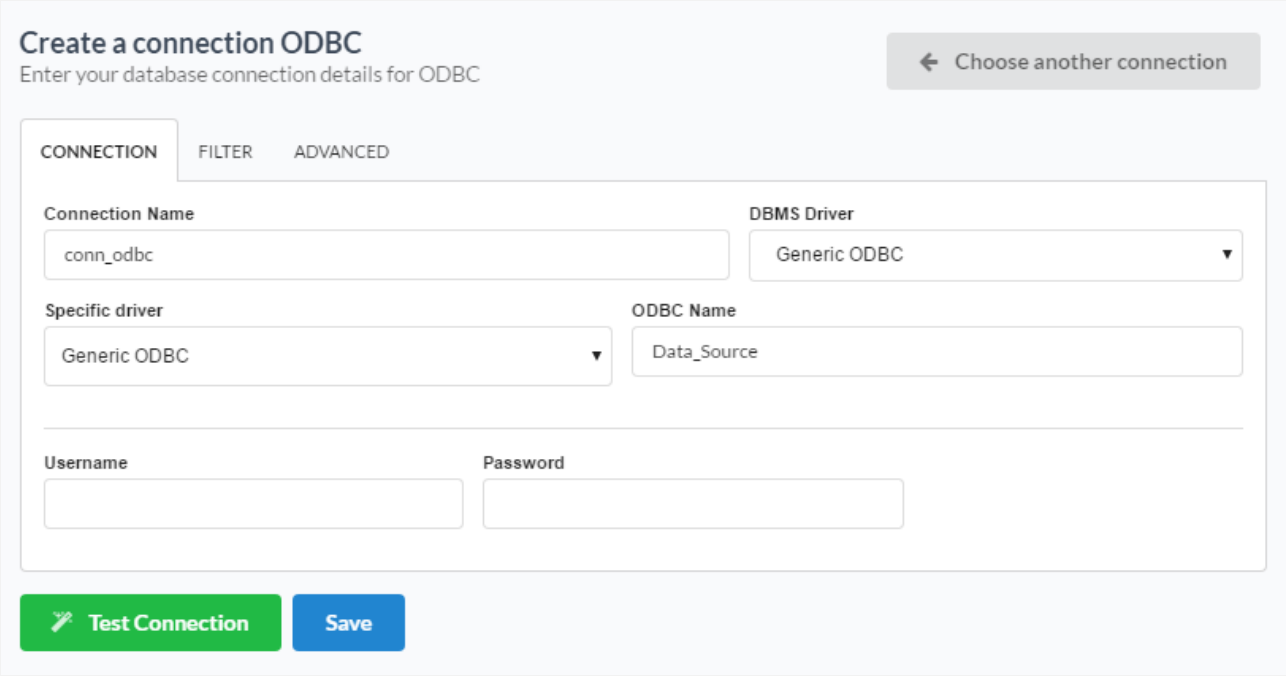
Creating a new connection

- After that, you will see a page with all the database connections that you can create. Select a “ODBC” connection



Selecting a Mac OS X database connection

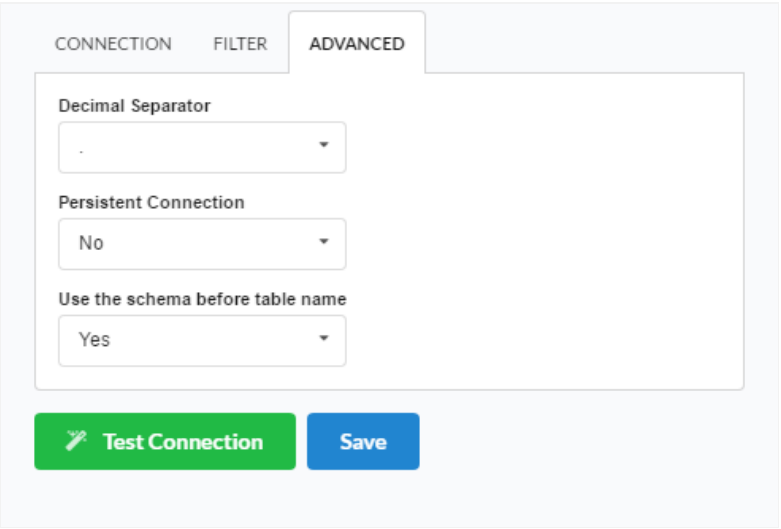
- **New connection** : You will place the information of the connection to your database here:



Connecting with Mac OS X database

- **Connection Name** : Defines the name of your new connection.
- **DBMS Driver** : Select the Generic ODBC.
- **Specific driver** : Defined as Generic ODBC, allows you to connect to specific Data Sources: DBF, FileMaker, Progress or Visual FoxPro.
- **ODBC Name** : On this option, you will inform the name of the Data Source that you have created.
- **Username** : Inform the username of the Mac OS X database, only if there are any.
- **Password** : Inform the password of the Mac OS X database, only if there are any.
- **Test Connection** : Displays a message of the status of the connection if successful or not.

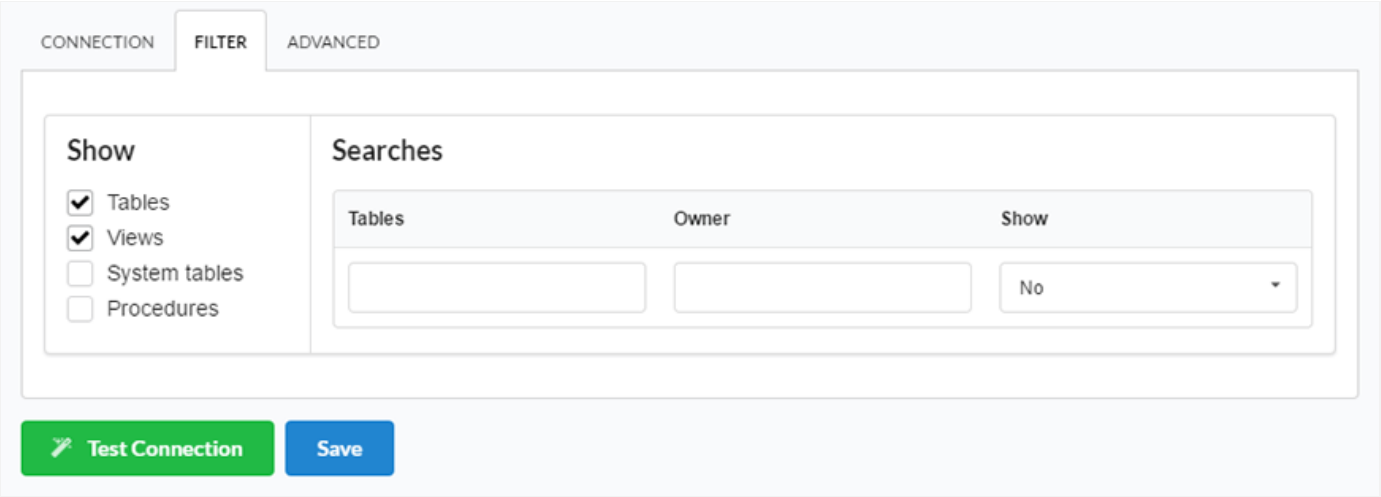
- For more options on connecting, click on the **Advanced** tab:



Advanced setup for the Mac OS X database

- **Decimal Separator** - Select the separator type, between a dot or a comma.
- **Persistent Connection** - Persistent Connection are connections that do not close when finishing a script.
- **Use the schema before table name** - Allows the use of schema before the name of the tables.

- **Filter** : Accessing this tab, we can setup which tables will be listed on this connection:



Filtering the Mac OS X database

- **Show** - Allows the display of the the filters on the tables, views, System Tables and procedures.
  - **Tables** - Allows to setup the display of Tables from your database.
  - **Views** - Allows to setup the display of Views from your database.
  - **System Tables** - Allows to setup the display of System Tables from your database.
  - **Procedures** - Allows to setup the display of Procedures from your database.
- **Searches** - Allows to define which tables and Owner will be displayed.
  - **Tables** - Allows to define a prefix (prefix%) or name of the tables for display.
  - **Owner** - Allows to define the owner of the tables for listing.
  - **Show** - Allows to define what will be displayed or not from the table an owner's setup.

Advanced

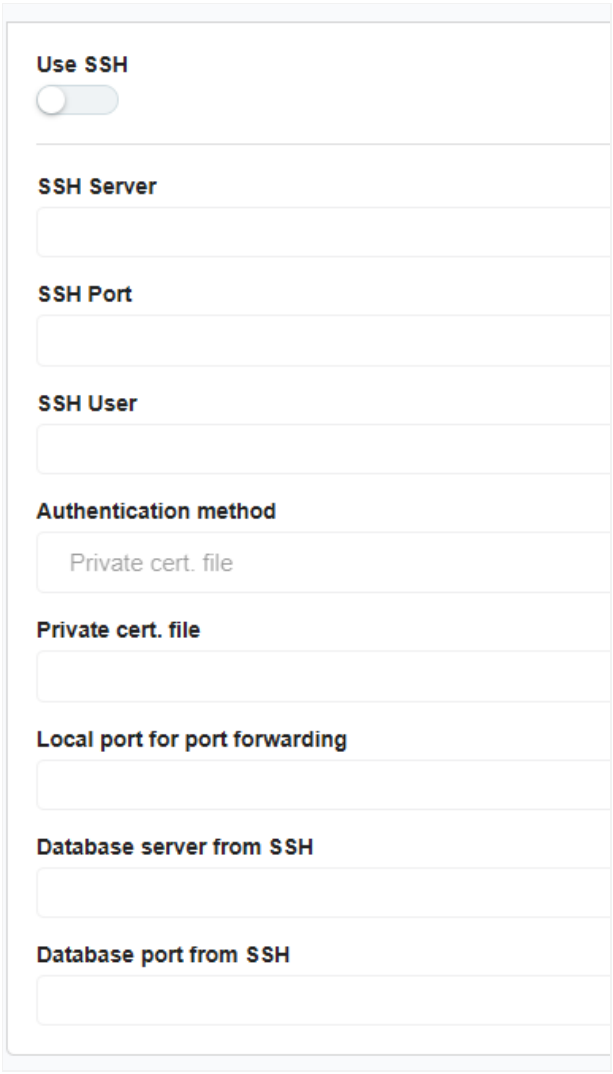
In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

![Advanced database connection configuration][conexao_avancado]

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the `.` dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

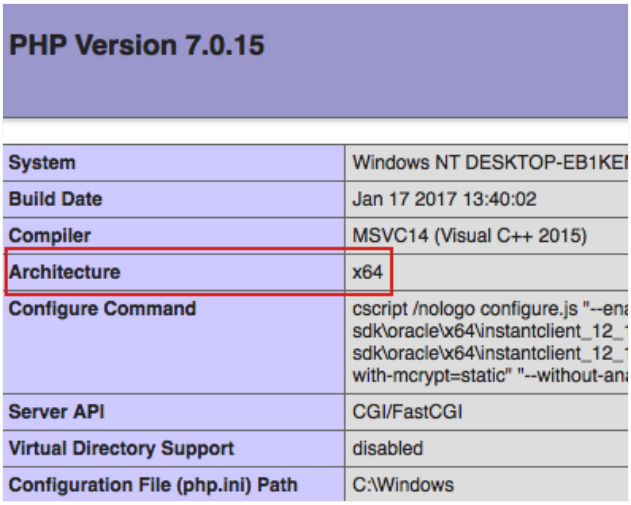
The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

.png

Progress

Windows

Before proceeding with this tutorial check the architecture of your PHP within the phpinfo (). Accessing the phpinfo of your scriptcase, for example **http://127.0.0.1:8092/scriptcase/info.php**, you will find the PHP architecture.



PHP Version 7.0.15

System	Windows NT DESKTOP-EB1KEI
Build Date	Jan 17 2017 13:40:02
Compiler	MSVC14 (Visual C++ 2015)
Architecture	x64
Configure Command	cscript /nologo configure.js "--ena sdk\oracle\x64\instantclient_12_ sdk\oracle\x64\instantclient_12_ with-mcrypt=static" "--without-ana
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	C:\Windows

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

If you are using Scriptcase automatic installer the PHP architecture will be the same as the installer you downloaded.

Progress OpenEdge ODBC

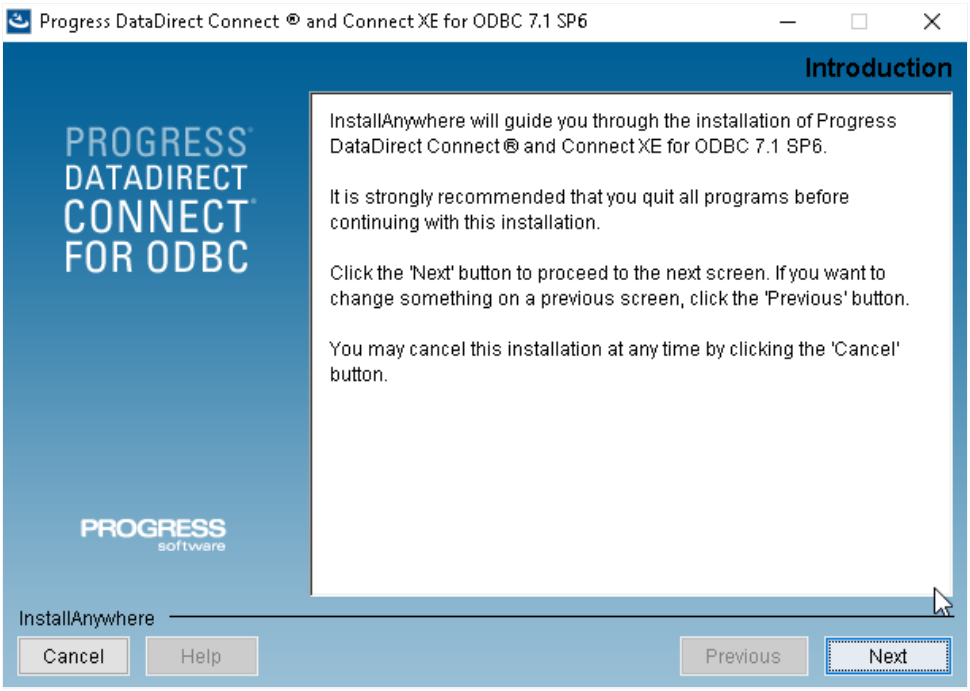
Setting up Progress connection using ODBC on Windows

Requirements

- The Progress Database needs to be installed.
- Have one or more tables created.
- Driver Progress® OpenEdge® ODBC Connector for Windows installed.

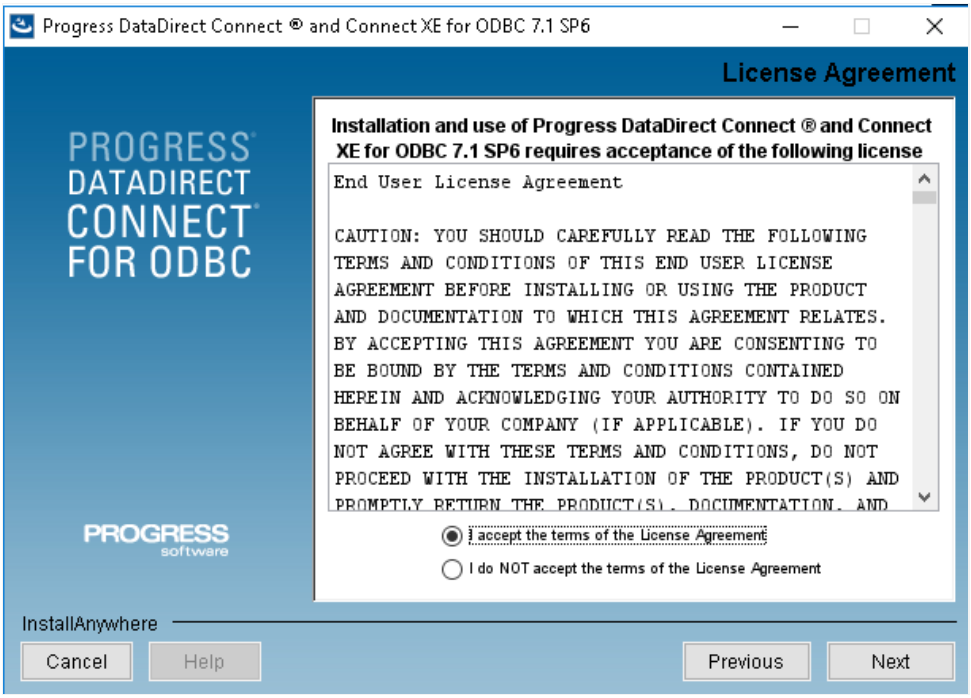
Enabling the extension

- First we must download the Progress® OpenEdge® ODBC Connector for Windows according to the architecture of your PHP.
  - To download the 64-bit engine click here
  - To download the 32-bit engine click here
- At the end of the download, follow the Progress OpenEdge ODBC Connector installation wizard.



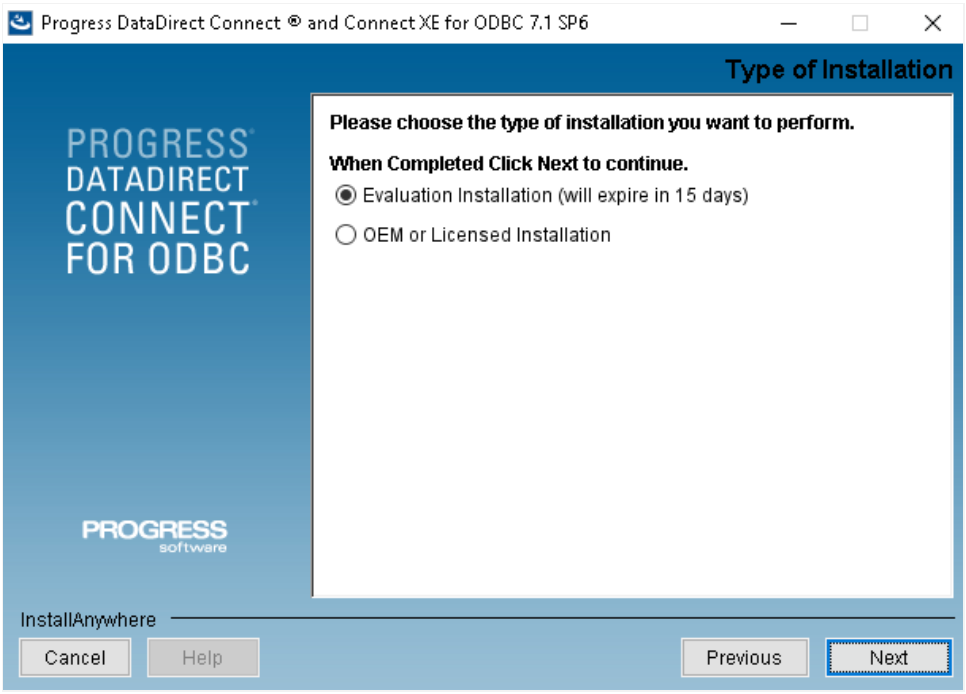
Installing the Progress OpenEdge ODBC

Connector.

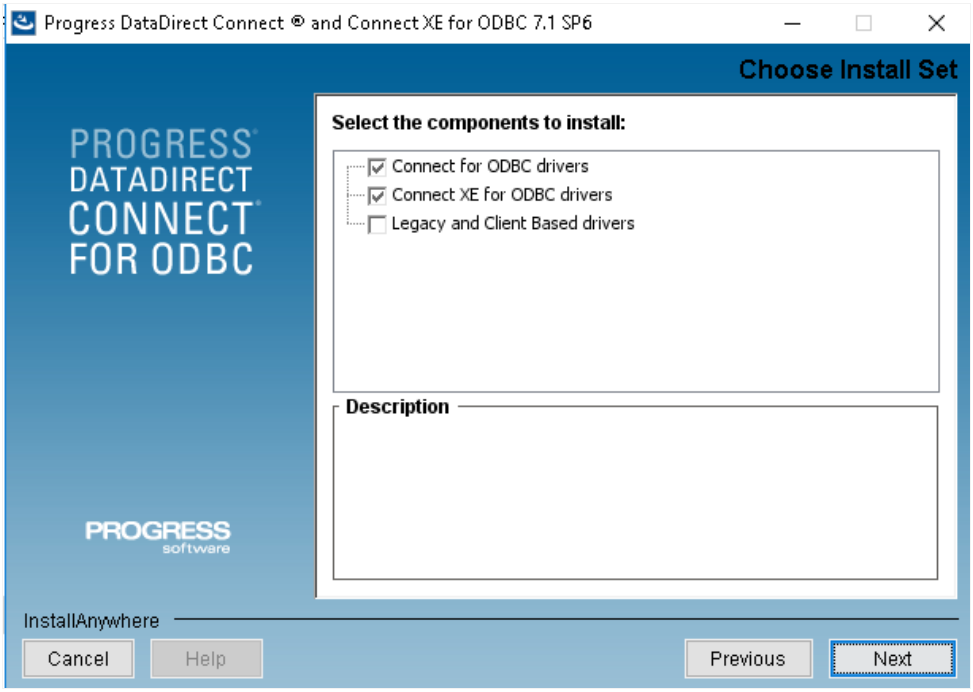


Installing the Progress OpenEdge ODBC

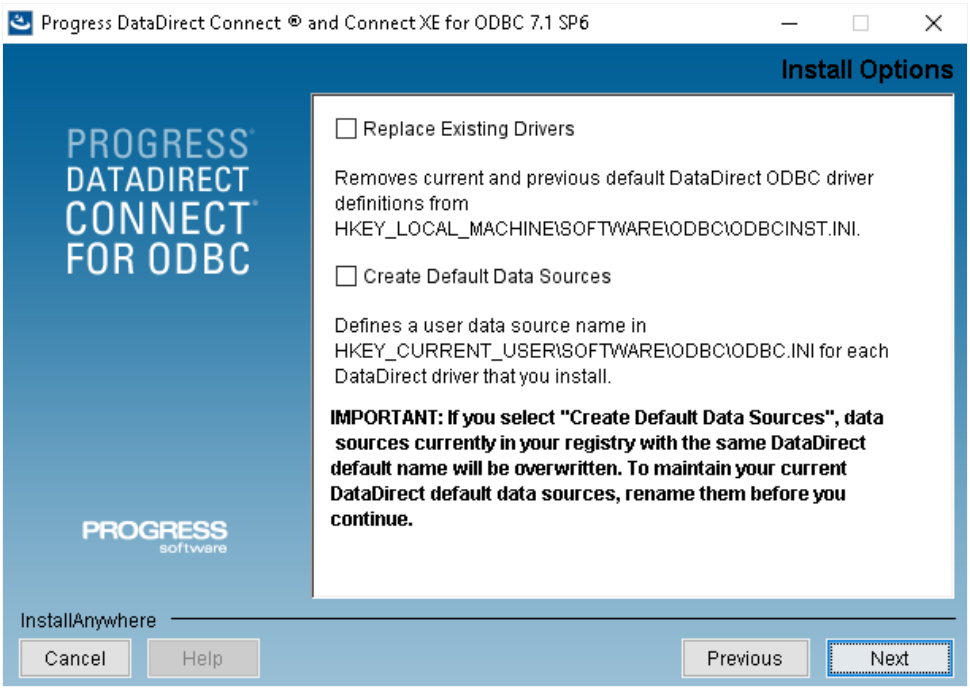
Connector.



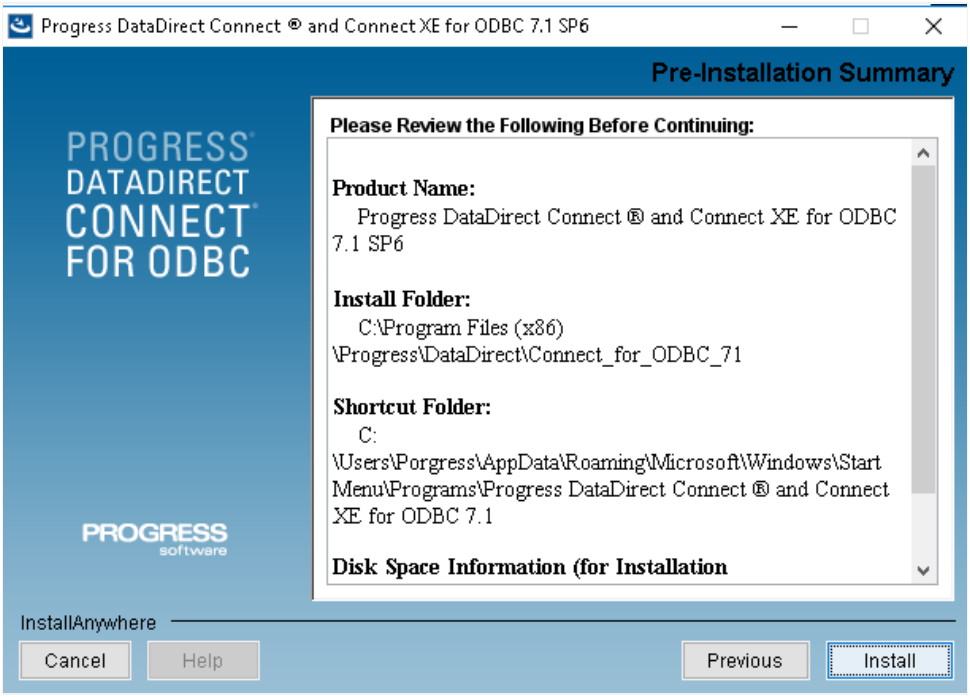
Installing the Progress OpenEdge ODBC Connector.



Installing the Progress OpenEdge ODBC Connector.



Installing the Progress OpenEdge ODBC Connector.

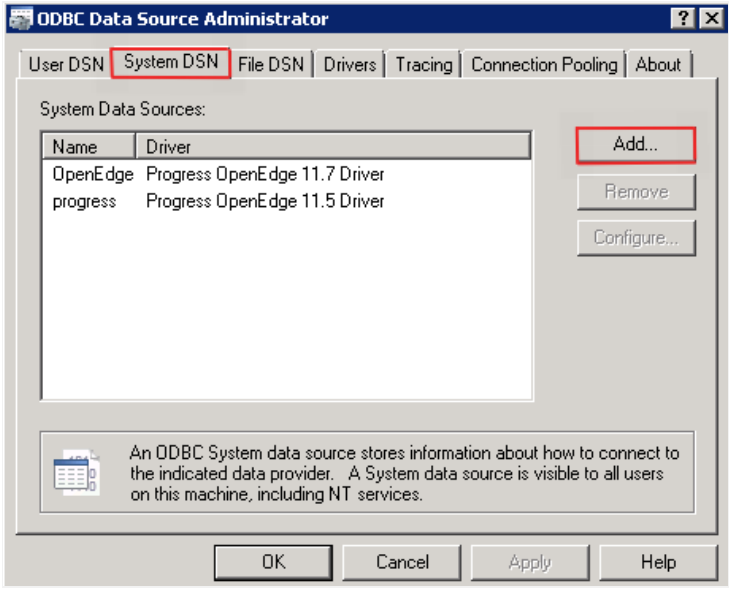


Installing the Progress OpenEdge ODBC Connector.

Installing the Progress OpenEdge ODBC Connector.

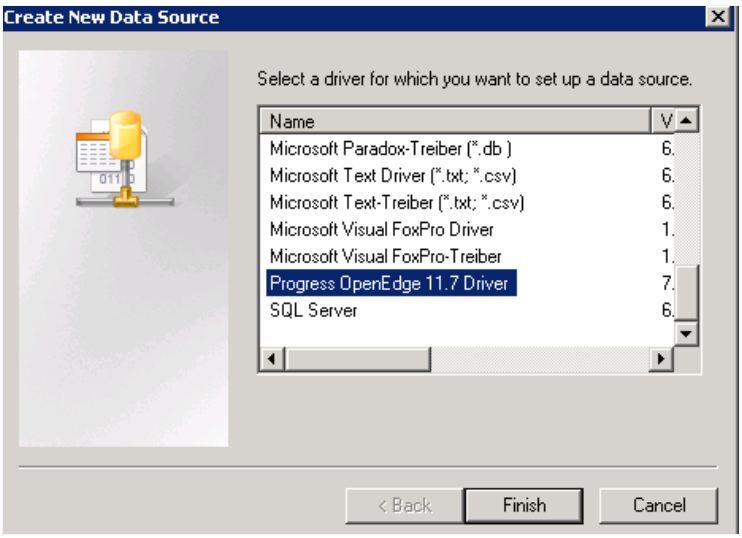
Creating a Data Source (ODBC)

- Go to Control Panel> Systems and Security> Administrative Tools> ODBC Data Sources (32 or 64 bits), according to your PHP architecture.
    - If you have a data source configured, proceed to Creating a Connection in ScriptCase.
- Select the System DNS tab and click Add.



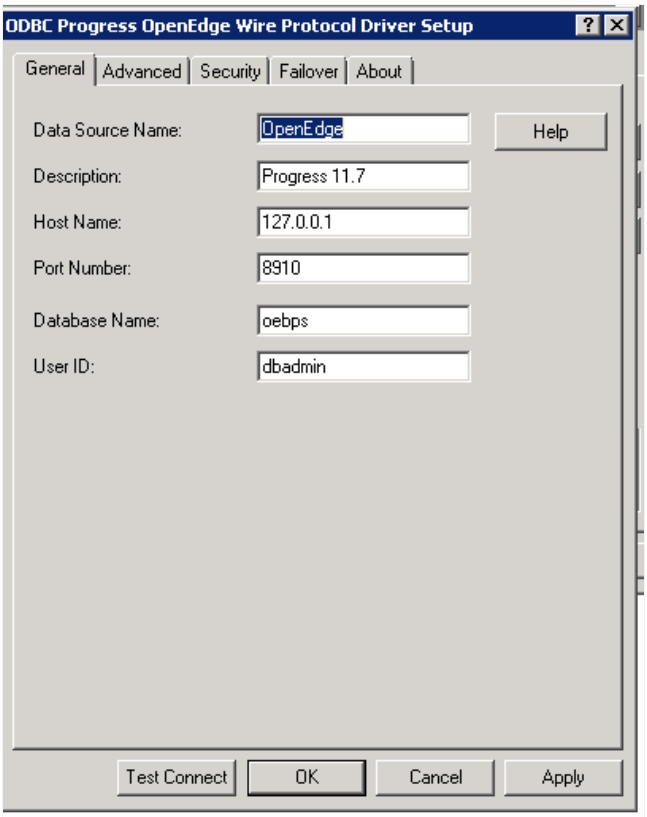
Accessing and Configuring the ODBC Data Source.

- Chose the following driver:
    - Progress OpenEdge 11.7 Driver



Accessing and Configuring the ODBC Data Source.

- After selecting the driver, the following screen will be displayed:
  - Six fields are required for data source configuration:
    - **Data Source Name**: Name of the data source that will be used when creating the connection in Scriptcase;
    - **Description**: A description for your driver;
    - **Host Name**: IP from where the database is installed;
    - **Port Number**: Port Number;
    - **Data Base Name**: Database name that will be used in the connection;
    - **User ID**: User name;



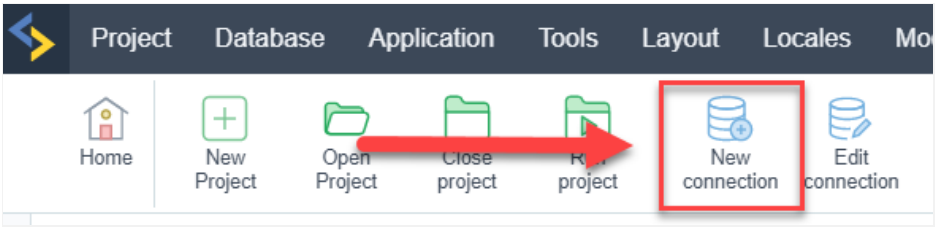
Creating the Data Source.

- After selecting the database, click OK to complete the configuration.
- Returning to the ODBC Data Source Administrator screen you can view the already configured connections.

Creating a Connection with ScriptCase

- Access ScriptCase using the URL http://127.0.0.1:8092/scriptcase

- After accessing ScriptCase, access or create a new project, click on the new connection icon or access “Database > New Connection”



Creating a new connection

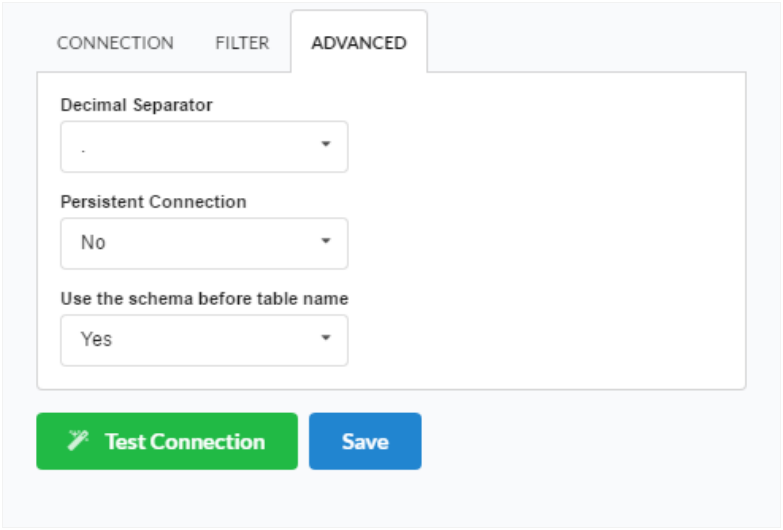
- After that, you will see a page with all the database connections that you can create. Select a “” connection

Selecting a Windows database connection

- **New connection** : You will place the information of the connection to your database here:

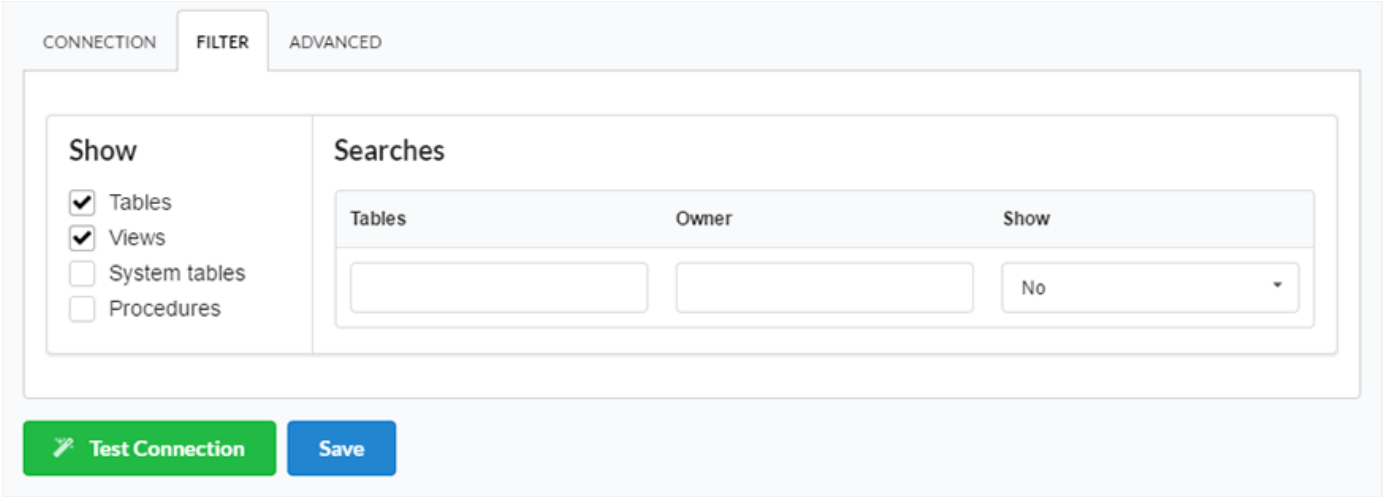
Connecting with Windows database

- For more options on connecting, click on the **Advanced** tab:



Advanced setup for the Windows database

- **Filter** : Accessing this tab, we can setup which tables will be listed on this connection:



Filtering the Windows database

- **Show** - Allows the display of the the filters on the tables, views, System Tables and procedures.
  - **Tables** - Allows to setup the display of Tables from your database.
  - **Views** - Allows to setup the display of Views from your database.
  - **System Tables** - Allows to setup the display of System Tables from your database.
  - **Procedures** - Allows to setup the display of Procedures from your database.
- **Searches** - Allows to define which tables and Owner will be displayed.
  - **Tables** - Allows to define a prefix (prefix%) or name of the tables for display.
  - **Owner** - Allows to define the owner of the tables for listing.
  - **Show** - Allows to define what will be displayed or not from the table an owner's setup.

Advanced

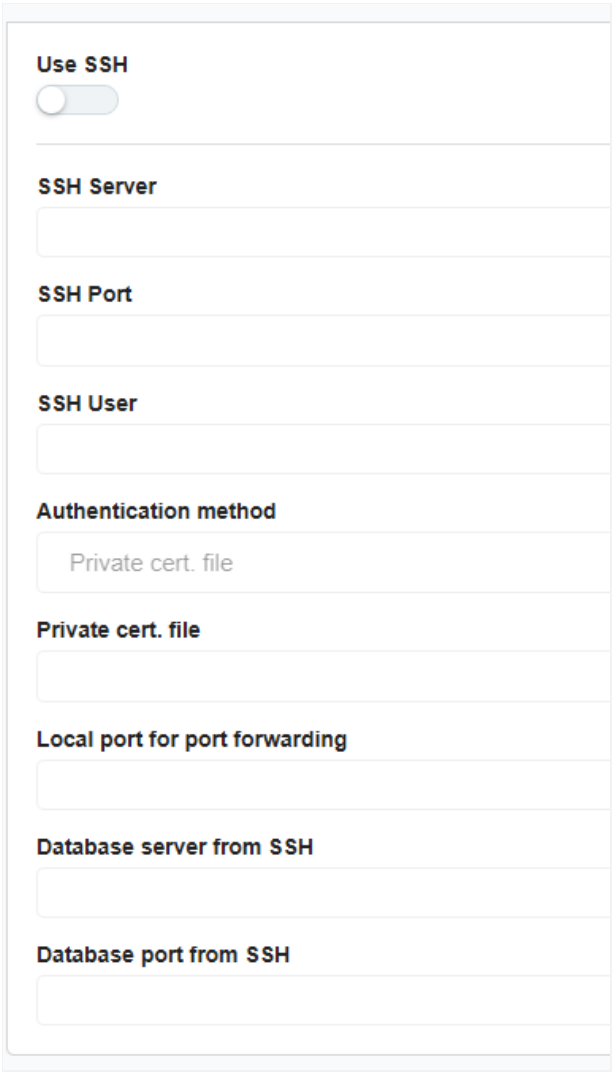
In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

![Advanced database connection configuration][conexao_avancado]

- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the `.` dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option .**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:



Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server's network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service's listening port.

.png

Microsoft Azure in Scriptcase

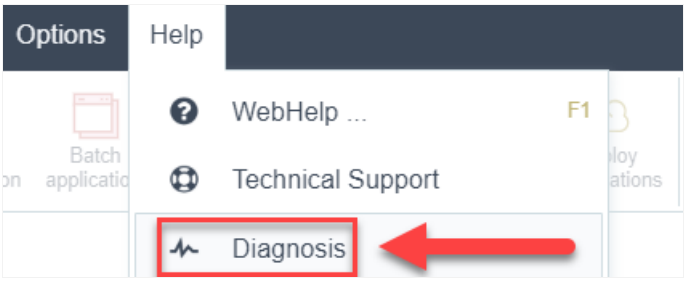
At Scriptcase, we have several banks (**MySQL, PostgreSQL and MSSQL Server**) and their respective drivers for connecting to Azure. You must choose the most suitable for your environment and need.

If you do not have an Azure database created from the ones available on Scriptcase, check below how to create a database:

- **Creation of Azure MySQL database:** click here
- **Creation of Azure PostgreSQL database:** click here
- **Creation of Azure MSSQL Server database:** click here

You can check your enabled drivers by accessing your diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.



- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `dominio.com/scriptcase/diagnosis.php`

Check below how to enable each database in Scriptcase according to your operating system:

MySQL

- **Windows:** click here
- **Linux:** click here
- **macOS:** click here

MariaDB

- **Windows:** click here
- **Linux:** click here
- **macOS:** click here

PostgreSQL

- **Windows:** click here
- **Linux:** click here
- **macOS:** click here

MSSQL Server

- **Windows:** click here
- **Linux:** click here
- **macOS:** click here

Enabling Azure MySQLi

In the Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be enabled manually in PHP.

Configuring and enabling the MySQLi on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to MySQL extensions: **mysqli** and **pdo_mysql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=mysqli
extension=pdo_mysql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection with Scriptcase

Check below which Driver you want to connect to.

- **MySQLi:** Click Here

Enabling Azure MySQL PDO

In the Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be enabled manually in PHP.

Configuring and enabling the MySQL PDO

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to MySQL extensions: **mysqli** and **pdo_mysql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=mysqli
extension=pdo_mysql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection with Scriptcase

Check below which Driver you want to connect to.

- **MySQL PDO:** Click here

Connection with Azure MySQL

If you do not have the **MySQLi** or **MySQL PDO** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MySQLi**: Click Here
- To enable the **MySQL PDO**: Click Here

Attention: If you do not have an Azure MySQL base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MySQLi or MySQL PDO**) and the Azure MySQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the database **Azure MySQL**.

Connection

Enter the parameters for connecting to your Azure MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MySQL Driver for connection. In this example, we use the **MySQL PDO** Driver.
- **SGDB Server**: Enter the domain of the Azure server where the database is installed.
- **EX:** serverdomain.mysql.database.azure.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Azure MySQL database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and __ improve the performance__ of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure MySQLi

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi .** If you are using your own pre-configured environment, MySQL extensions must be enabled manually in PHP.

Configuring and enabling the MySQLi on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install MySQL extensions. See below for how to proceed.

```
sudo apt-get install php8.1-mysql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **MySQLi:** Clique aqui

Enabling Azure MySQL PDO

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be enabled manually in PHP.

Configuring and enabling the MySQL PDO on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install MySQL extensions. See below for how to proceed.

```
sudo apt-get install php8.1-mysql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **MySQL PDO:** Click Here

Connection with Azure MySQL

If you do not have the **MySQLi** or **MySQL PDO** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MySQLi**: Click Here
- To enable the **MySQL PDO**: Click Here

Attention: If you do not have an Azure MySQL base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MySQLi or MySQL PDO**) and the Azure MySQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the database **Azure MySQL**.

Connection

Enter the parameters for connecting to your Azure MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MySQL Driver for connection. In this example, we use the **MySQL PDO** Driver.
- **SGDB Server**: Enter the domain of the Azure server where the database is installed.
- **EX:** serverdomain.mysql.database.azure.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Azure MySQL database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and __ improve the performance__ of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure MySQLi

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi .** If you are using your own pre-configured environment, MySQL extensions must be enabled manually in PHP.

Configuring and enabling the MySQLi on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **MySQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 in macOS: Click here

Connection to Scriptcase

- **MySQLi:** Click here

Enabling Azure MySQL PDO

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi .** If you are using your own pre-configured environment, MySQL extensions must be enabled manually in PHP.

Configuring and enabling the MySQL PDO on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **MySQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 in macOS: Click Here

Connection to Scriptcase

- **MySQL PDO:** Click Here

Connection with Azure MySQL

If you do not have the **MySQLi** or **MySQL PDO** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MySQLi**: Click Here
- To enable the **MySQL PDO**: Click Here

Attention: If you do not have an Azure MySQL base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MySQLi or MySQL PDO**) and the Azure MySQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the database **Azure MySQL**.

Connection

Enter the parameters for connecting to your Azure MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MySQL Driver for connection. In this example, we use the **MySQL PDO** Driver.
- **SGDB Server**: Enter the domain of the Azure server where the database is installed.
- **EX:** serverdomain.mysql.database.azure.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Azure MySQL database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and __ improve the performance__ of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure PostgreSQL PDO

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Azure PostgreSQL PDO on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines so that the extension is used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Enabling Azure PostgreSQL 7 or Above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Azure PostgreSQL 7 on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines so that the extension is used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Enabling Azure PostgreSQL 6.4 or Above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Azure PostgreSQL 6.4 on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines so that the extension is used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Enabling Azure PostgreSQL 6.3 ou Below

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Azure PostgreSQL 6.3 on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines so that the extension is used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Connection to Azure PostgreSQL

If you do not have the **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **PostgreSQL PDO:** Click Here
- To enable the **PostgreSQL 7 or Above:** Click Here
- To enable the **PostgreSQL 6.4 or Above:** Click Here
- To enable the **PostgreSQL 6.3 or Below:** Click Here

ATTENTION: If you do not have an Azure PostegreSQL base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.**) and the Azure PostgreSQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the **Azure Postgres** database.

Connection

Enter the parameters for connecting to your Azure PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PostgreSQL PDO** Driver.
- **SGDB server**: Enter the domain of the Azure server where the database is installed.
- **EX:** domain-postgres.postgres.database.azure.com
- **Port**: Enter the port for connection. By default, the defined port is **5432**.
- **Schema**: Enter the schema for connecting to the database.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the

parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate):

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key):

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate):

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed

- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filter

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Display:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure PostgreSQL PDO

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Azure PostgreSQL PDO on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Enabling Azure PostgreSQL 7 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Azure PostgreSQL 7 or above on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL 7 or Above:** Click Here

Enabling Azure PostgreSQL 6.4 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Azure PostgreSQL 6.4 or above on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL 6.4 or Above:** Click Here

Enabling Azure PostgreSQL 6.3 or below

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Azure PostgreSQL 6.3 or below on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL 6.3 or Below:** Click Here

Connection to Azure PostgreSQL

If you do not have the **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **PostgreSQL PDO:** Click Here
- To enable the **PostgreSQL 7 or Above:** Click Here
- To enable the **PostgreSQL 6.4 or Above:** Click Here
- To enable the **PostgreSQL 6.3 or Below:** Click Here

ATTENTION: If you do not have an Azure PostegreSQL base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.**) and the Azure PostgreSQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the **Azure Postgres** database.

Connection

Enter the parameters for connecting to your Azure PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PostgreSQL PDO** Driver.
- **SGDB server**: Enter the domain of the Azure server where the database is installed.
- **EX:** domain-postgres.postgres.database.azure.com
- **Port**: Enter the port for connection. By default, the defined port is **5432**.
- **Schema**: Enter the schema for connecting to the database.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the

parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate):

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key):

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate):

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed

- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filter

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Display:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure PostgreSQL PDO

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling PostgreSQL on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click Here

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Enabling Enabling Azure PostgreSQL 7 or Above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling PostgreSQL on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click Here

Connection to Scriptcase

- **PostgreSQL 7 or Above:** Click Here

Enabling Azure PostgreSQL 6.4 or Above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling PostgreSQL 6.4 or Above on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click Here

Conexão com o Scriptcase

- **PostgreSQL 6.4 or Above:** Click Here

Enabling Azure PostgreSQL 6.3 or below

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling PostgreSQL 6.3 or below on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click Here

Connection to Scriptcase

- **PostgreSQL 6.3 or Below:** Click Here

Connection to Azure PostgreSQL

If you do not have the **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **PostgreSQL PDO:** Click Here
- To enable the **PostgreSQL 7 or Above:** Click Here
- To enable the **PostgreSQL 6.4 or Above:** Click Here
- To enable the **PostgreSQL 6.3 or Below:** Click Here

ATTENTION: If you do not have an Azure PostegreSQL base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.**) and the Azure PostgreSQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the **Azure Postgres** database.

Connection

Enter the parameters for connecting to your Azure PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PostgreSQL PDO** Driver.
- **SGDB server**: Enter the domain of the Azure server where the database is installed.
- **EX:** domain-postgres.postgres.database.azure.com
- **Port**: Enter the port for connection. By default, the defined port is **5432**.
- **Schema**: Enter the schema for connecting to the database.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the

parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate):

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key):

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate):

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed

- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filter

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Display:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improve the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure MSSQL Server SRV

At Scriptcase, we have the following drivers available for connection to Azure MSSQL Server: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC .** If you are using your own pre-configured environment, the MSSQL Server extensions must be enabled manually in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 11 For SQL Server according to your PHP architecture .**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Listed below are the necessary files for the drivers to be enabled.

Required files:

x64

- Microsoft ODBC Driver 11 For SQL Server **(x64)**: Click Here

x86

- Microsoft ODBC Driver 11 For SQL Server **(x86)**: Click Here

Other Files

- PHP extensions **(php_pdo_sqlsrv_73_nts_x64 and php_sqlsrv_73_nts_x64)**: Click Here

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling Azure MSSQL Server SRV on Windows

IMPORTANT: If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, requiring only the installation of the client. **Proceed to item 3.**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64**. See the example below:

```
extension=php_pdo_sqlsrv_73_nts_x64
```

extension=php_sqlsrv_73_nts_x64

3 - After downloading the Microsoft ODBC Driver 11 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **MSSQL Server NATIVE SRV:** Click here

Enabling Azure MSSQL Server SRV PDO

At Scriptcase, we have the following drivers available for connection to Azure MSSQL Server: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC .** If you are using your own pre-configured environment, the MSSQL Server extensions must be enabled manually in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 11 For SQL Server according to your PHP architecture .**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Listed below are the necessary files for the drivers to be enabled.

Required files:

x64

- Microsoft ODBC Driver 11 For SQL Server **(x64)**: Click Here

x86

- Microsoft ODBC Driver 11 For SQL Server **(x86)**: Click Here

Other Files

- PHP extensions **(php_pdo_sqlsrv_73_nts_x64 and php_sqlsrv_73_nts_x64)**: Click Here

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling Azure MSSQL Server SRV PDO on Windows

IMPORTANT: If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, requiring only the installation of the client. **Proceed to item 3.**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64**. See the example below:

```
extension=php_pdo_sqlsrv_73_nts_x64
extension=php_sqlsrv_73_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 11 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **MSSQL Server NATIVE SRV PDO:** Click here

Enabling Azure MSSQL Server ODBC

At Scriptcase, we have the following drivers available for connection to Azure MSSQL Server: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC .** If you are using your own pre-configured environment, the MSSQL Server extensions must be enabled manually in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 11 For SQL Server according to your PHP architecture .**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Listed below are the necessary files for the drivers to be enabled.

Required files:

x64

- Microsoft ODBC Driver 11 For SQL Server **(x64)**: Click Here

x86

- Microsoft ODBC Driver 11 For SQL Server **(x86)**: Click Here

Other Files

- PHP extensions **(php_pdo_sqlsrv_73_nts_x64 and php_sqlsrv_73_nts_x64)**: Click Here

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling Azure MSSQL Server ODBC on Windows

IMPORTANT: If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, requiring only the installation of the client. **Proceed to item 3.**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64**. See the example below:

```
extension=php_pdo_sqlsrv_73_nts_x64
extension=php_sqlsrv_73_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 11 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **MSSQL Server ODBC:** Click here

Connection to Azure MSSQL

If you do not have the **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MSSQL Server NATIVE SRV**: Click here
- To enable the **MSSQL Server NATIVE SRV PDO**: Click here

  ATTENTION: If you do not have an Azure MSSQL Server base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**MSSQL Server NATIVE SRV or MSSQL Server NATIVE SRV PDO**) and the Azure MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the **Azure SQL** database.

Connection

Enter the parameters for connecting to your Azure MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MSSQL Server Driver to connect. In this example, we use the **MSSQL Server NATIVE SRV PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Azure server where the database is installed.
- **EX:** domain-sql.database.windows.net
- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to Azure MSSQL ODBC

If you do not have the **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MSSQL Server ODBC**: Click here

  ATTENTION: If you do not have an Azure MSSQL Server base created, check how to create it by clicking here.

Creating the DSN for connection

To connect to the **MSSQL Server ODBC** Driver, you need to configure a System DSN. Follow the steps described below to perform this configuration.

> **ATTENTION**: ODBC must be created on the same server where ScriptCase is installed.

1 - Access your **ODBC Data Source Manager** and select according to your architecture. We will select the x64 version:

2 - When entering the ODBC Data Sources Administrator, select the **DSN System** tab and click on *Add* to create your connection to your Database.

3 - After that, it is necessary to select the Driver to connect to the MSSQL Server. Select the Driver: **ODBC Driver 11 for SQL Server.**

4 - Now, you need to define the name of the DSN, the description and the database connection server.

- **Name:** Enter the name that the DSN will have so that you can use it in Scriptcase.
- **Description:** Add a description for the DSN.
- **Server:** inform the server, port and database instance that you will connect to.

4.1 - After informing the data, click on the **Next** button to inform the user name and password required for connection.

- **Login ID:** Inform the user to authenticate with the database.
- **Password:** Enter the corresponding password to authenticate with the informed user.

4.2 - Now, click **Next** to select the default database.

To finish, click **Next** and finally, click **Finish**.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**MSSQL Server ODBC**) and the Azure MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the **Azure SQL** database.

Connection

Enter the parameters for connecting to your Azure MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MSSQL Server Driver to connect. In this example, we use the **MSSQL Server NATIVE SRV PDO** Driver.
- **Server/Host (Name or IP)**: Enter the Data Source Name of the Azure server.
- **EX:** mssql
- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure PDO DBLIB

In Scriptcase installed on Linux, we have the following driver available for connection to MSSQL Server: **PDO DBLIB.** If you are using your own pre-configured environment, MSSQL Server extensions must be enabled manually in PHP.

Configuring and enabling Azure PDO DBLIB on Linux

> **IMPORTANT:** If you are using the Scriptcase automatic installer, **the extensions are already enabled in PHP**, all that is missing is the configuration of the freetds file. **The procedure below is only for manual installations of Scriptcase. To see how to configure freetds in the automatic installation, scroll down to the configuration section by clicking here**

1 - If you do not have the extension enabled you can install the extension with this command in the terminal:

```
sudo apt-get install php8.1-pdo-dblib
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

Install the basic packages using the terminal for this procedure:

Ubuntu\Debian	RHEL\CentOS
sudo apt-get install unixodbc unixodbc-dev gcc nano wget make	sudo yum install unixODBC unixODBC-devel gcc nano wget make

- Download and Unzip the freeTDS:

```
wget https://www.freetds.org/files/stable/freetds-0.95.95.tar.gz

tar -zxf freetds-0.95.95.tar.gz
```

- Access the freeTDS directory and run the following command to compile and install FreeTDS:

```
sudo ./configure --with-tdsver=7.4 --with-unixodbc=/usr --disable-libiconv --disable-static --disable-threadsafe --enable-msdblib --disable-sspi --with-gnu-ld --enable-sybase-compat && make && make install
```

- Edit the FreeTDS configuration file:

Ubuntu\Debian	RHEL\CentOS
sudo nano/usr/local/etc/freetds.conf	sudo vim /usr/local/freetds/freetds.conf

- Add your MSSQL Server connection informations, like this example:

  ```
  [MSSQLServer] host = 192.18.72.03
  port = 1433
  tds version = 7.2
  ```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Setting up freetds - automatic installation

1 - Edit the FreeTDS configuration file:

Automatic installation directory
sudo nano /opt/Scriptcase/v9-php8/components/drivers/freetds/etc/freetds.conf

2 - Add your MSSQL Server connection informations, like in the example:

```
[MSSQLServer] host = 192.18.72.03
port = 1433
```

tds version = 7.4

3 - Restart Apache service

sudo service apachesc9php81 restart

Connection to Scriptcase

- **PDO DBLIB:** Clique aqui

Connection to Azure MSSQL

If you do not have the **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **PDO DBLIB**: Click here

  ATTENTION: If you do not have an Azure MSSQL Server base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**MSSQL Server NATIVE SRV or MSSQL Server NATIVE SRV PDO**) and the Azure MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the **Azure SQL** database.

Connection

Enter the parameters for connecting to your Azure MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MSSQL Server Driver to connect. In this example, we use the **PDO DBLIB** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Azure server where the database is installed.
- **EX:** domain-sql.database.windows.net
- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

- **Encrypt**

  This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

- **trustservercertificate**

  Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

  If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

- **trustStore**

  The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

  When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

- **trustStorePassword**

  The password used to verify the integrity of trustStore data.

  If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

- **hostnameInCertificate**

  The hostname to use to validate the SQL Server TLS/SSL certificate.

  This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

  Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure PDO DBLIB

In Scriptcase installed on macOs, we have the following driver available for connection to MSSQL Server: **PDO DBLIB .** If you are using your own pre-configured environment, MSSQL Server extensions must be enabled manually in PHP.

Configuring and enabling the DBLIB PDO on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **PDO DBLIB is already enabled in PHP**, requiring only the connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click here

Connection to Scriptcase

Check below which Driver you want to connect to.

- **PDO DBLIB:** Click here

Connection to Azure MSSQL

If you do not have the **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **PDO DBLIB**: Click here

  ATTENTION: If you do not have an Azure MSSQL Server base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**MSSQL Server NATIVE SRV or MSSQL Server NATIVE SRV PDO**) and the Azure MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the **Azure SQL** database.

Connection

Enter the parameters for connecting to your Azure MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MSSQL Server Driver to connect. In this example, we use the **PDO DBLIB** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Azure server where the database is installed.
- **EX:** domain-sql.database.windows.net
- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty .**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator .**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure MariaDB PDO

In the Scriptcase, we have the following driver available for connection to MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extension must be enabled manually in PHP.

Configuring and enabling the MariaDB PDO

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for a line referring to MariaDB extension: **pdo_mysql** and remove the ; from the beginning of the line for the extension to be used by PHP. See the example below:

```
extension=pdo_mysql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php81 or Apache2.4 service and right click on this service, then **Restart**.

Connection with Scriptcase

Check below which Driver you want to connect to.

- **MariaDB PDO:** Click here

Connection with Azure MariaDB

If you do not have the **MariaDB PDO** driver enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MariaDB PDO**: Click Here

  Attention: If you do not have an Azure MariaDB base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MariaDB PDO**) and the Azure MariaDB database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the database **Azure MariaDB**.

Connection

Enter the parameters for connecting to your Azure MariaDB database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MariaDB Driver for connection. In this example, we use the **MariaDB PDO** Driver.
- **SGDB Server**: Enter the domain of the Azure server where the database is installed.
- **EX:** serverdomain.mariadb.database.azure.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Azure MariaDB database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and __ improve the performance__ of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure MariaDB PDO

At Scriptcase, we have the following driver available for connection to MariaDB: **mariaDB PDO.** If you are using your own pre-configured environment, MariaDB extensions must be enabled manually in PHP.

Configuring and enabling the MariaDB PDO on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install MariaDB extensions. See below for how to proceed.

```
sudo apt-get install php8.1-mysql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **MariaDB PDO:** Click Here

Connection with Azure MariaDB

If you do not have the **MariaDB PDO** driver enabled, check below our documentation on how to enable in Scriptcase for connection.

- To enable the **MariaDB PDO**: Click Here

  Attention: If you do not have an Azure MariaDB base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MariaDB PDO**) and the Azure MariaDB database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the database **Azure MariaDB**.

Connection

Enter the parameters for connecting to your Azure MariaDB database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MariaDB Driver for connection. In this example, we use the **MariaDB PDO** Driver.
- **SGDB Server**: Enter the domain of the Azure server where the database is installed.
- **EX:** serverdomain.mariadb.database.azure.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Azure MariaDB database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and __ improve the performance__ of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Azure MariaDB PDO

At Scriptcase, we have the following driver available for connection to MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extension must be enabled manually in PHP.

Configuring and enabling the MariaDB PDO on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **MariaDB is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 in macOS: Click Here

Connection to Scriptcase

- **MariaDB PDO:** Click Here

Connection with Azure MariaDB

If you do not have the _**MariaDB PDO** driver enabled, check below our documentation on how to enable in Scriptcase for connection.

- To enable the **MariaDB PDO**: Click Here

  Attention: If you do not have an Azure MariaDB base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MariaDB PDO**) and the Azure MariaDB database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Microsoft Azure** connection.

4 - And choose the database **Azure MariaDB**.

Connection

Enter the parameters for connecting to your Azure MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MariaDB Driver for connection. In this example, we use the **MariaDB PDO** Driver.
- **SGDB Server**: Enter the domain of the Azure server where the database is installed.
- **EX:** serverdomain.mariadb.database.azure.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Azure MariaDB database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and __ improve the performance__ of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Amazon RDS in Scriptcase

At Scriptcase, we have several banks (**MySQL, PostgreSQL, MSSQL Server and Oracle**) and their respective drivers for connection with Amazon RDS. You must choose the most suitable for your environment and need.

If you do not have an Amazon bank created from the ones available on Scriptcase, check below how to create a database:

- **Creation of Amazon RDS database:** Click here

You can check your enabled drivers by accessing your diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `dominio.com/scriptcase/diagnosis.php`

Check below how to enable each database in Scriptcase according to your operating system:

MySQL

- **Windows:** Click here
- **Linux:** Click here
- **macOS:** Click here

MariaDB

- **Windows:** Click here
- **Linux:** Click here
- **macOS:** Click here

PostgreSQL

- **Windows:** Click here
- **Linux:** Click here
- **macOS:** Click here

MSSQL Server

- **Windows:** Click here
- **Linux:** Click here
- **macOS:** Click here

Oracle

- **Windows:** Click here
- **Linux:** Click here
- **macOS:** Click here

Enabling Amazon RDS MySQLi

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi .** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling Amazon RDS MySQLi on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

1 - In the php.ini file, located in C:\php, look for lines referring to MySQL extensions: **mysqli** and **pdo_mysql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=mysqli
extension=pdo_mysql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **MySQLi:** Click Here

Questions or Connection Problems?

Enabling Amazon RDS MySQL PDO

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi .** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling Amazon RDS MySQL PDO on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

1 - In the php.ini file, located in C:\php, look for lines referring to MySQL extensions: **mysqli** and **pdo_mysql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=mysqli
extension=pdo_mysql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **MySQL PDO:** Click Here

Questions or Connection Problems?

Connection to Amazon RDS MySQL

If you do not have the **MySQLi** or **MySQL PDO** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MySQLi**: Click Here
- To enable the **MySQL PDO**: Click Here

CAUTION: If you don't have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MySQLi or MySQL PDO**) and the Amazon RDS MySQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS MySQL** database.

Connection

Enter the parameters for connecting to your Amazon RDS MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MySQL Driver for connection. In this example, we use the **MySQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-mysql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Amazon RDS MySQL database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option .**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed

before the table names.
- **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS MySQLi

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling MySQLi on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install MySQL extensions. See below for how to proceed.

```
sudo apt-get install php8.1-mysql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php sudo apt-get update
```

2 - Restart the Apache service through the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **MySQLi:** Click Here

Enabling Amazon RDS MySQL PDO

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling MySQL PDO on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install MySQL extensions. See below for how to proceed.

```
sudo apt-get install php8.1-mysql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php sudo apt-get update
```

2 - Restart the Apache service through the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **MySQL PDO:** Click Here

Connection to Amazon RDS MySQL

If you do not have the **MySQLi** or **MySQL PDO** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MySQLi**: Click Here
- To enable the **MySQL PDO**: Click Here

CAUTION: If you don't have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MySQLi or MySQL PDO**) and the Amazon RDS MySQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS MySQL** database.

Connection

Enter the parameters for connecting to your Amazon RDS MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MySQL Driver for connection. In this example, we use the **MySQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-mysql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Amazon RDS MySQL database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option .**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed

before the table names.
- **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS MySQLi

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling MySQLi on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

If you have installed PHP 8.1 through our documentation using homebrew, **MySQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 no macOS: Click Here

Connection to Scriptcase

- **MySQLi:** Click Here

Questions or Connection Problems?

Enabling Amazon RDS MySQL PDO

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling MySQL PDO on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

If you have installed PHP 8.1 through our documentation using homebrew, **MySQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 no macOS: Click Here

Connection to Scriptcase

- **MySQL PDO:** Click Here

Questions or Connection Problems?

Connection to Amazon RDS MySQL

If you do not have the **MySQLi** or **MySQL PDO** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MySQLi**: Click Here
- To enable the **MySQL PDO**: Click Here

CAUTION: If you don't have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MySQLi or MySQL PDO**) and the Amazon RDS MySQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS MySQL** database.

Connection

Enter the parameters for connecting to your Amazon RDS MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MySQL Driver for connection. In this example, we use the **MySQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-mysql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Amazon RDS MySQL database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option .**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed

before the table names.
- **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS PostgreSQL PDO

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Amazon RDS PostgreSQL PDO on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Enabling Amazon RDS PostgreSQL 7 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Amazon RDS PostgreSQL 7 or above on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL 7 or above:** Click Here

Enabling Amazon RDS PostgreSQL 6.4 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Amazon RDS PostgreSQL 6.4 or above on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL 6.4 or above:** Click Here

Enabling Amazon RDS PostgreSQL 6.3 or below

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Amazon RDS PostgreSQL 6.3 or below on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL 6.3 or below:** Click Here

Connection to Amazon RDS PostgreSQL

If you do not have the **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **PostgreSQL PDO:** Click Here
- To enable the **PostgreSQL 7 or above:** Click Here
- To enable the **PostgreSQL 6.4 or above:** Click Here
- To enable the **PostgreSQL 6.3 or below:** Click Here

ATTENTION: If you don't have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Driver enabled (**PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.**) and the Amazon RDS PostgreSQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS Postgres** database.

Connection

Enter the parameters for connecting to your Amazon RDS PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PostgreSQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-postgres.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **5432**.
- **Eschema**: Enter the schema for connection to the database.
- **Database NAme**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.

- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate):

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key):

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate):

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS PostgreSQL PDO

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Amazon RDS PostgreSQL PDO on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL PDO:** Click here

Enabling Amazon RDS PostgreSQL 7 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Amazon RDS PostgreSQL 7 or above on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL 7 or above:** Click here

Enabling Amazon RDS PostgreSQL 6.4 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Amazon RDS PostgreSQL 6.4 or above on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL 6.4 or above:** Click here

Enabling Amazon RDS PostgreSQL 6.3 or below

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Amazon RDS PostgreSQL 6.3 or below on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL 6.3 or below:** Click here

Connection to Amazon RDS PostgreSQL

If you do not have the **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **PostgreSQL PDO:** Click Here
- To enable the **PostgreSQL 7 or above:** Click Here
- To enable the **PostgreSQL 6.4 or above:** Click Here
- To enable the **PostgreSQL 6.3 or below:** Click Here

ATTENTION: If you don't have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Driver enabled (**PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.**) and the Amazon RDS PostgreSQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS Postgres** database.

Connection

Enter the parameters for connecting to your Amazon RDS PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PostgreSQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-postgres.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **5432**.
- **Eschema**: Enter the schema for connection to the database.
- **Database NAme**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.

- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate):

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key):

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate):

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS PostgreSQL PDO

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling PostgreSQL PDO on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click Here

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Enabling Amazon RDS PostgreSQL 7 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling PostgreSQL 7 or above on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click Here

Connection to Scriptcase

- **PostgreSQL 7 or above:** Click Here

Enabling Amazon RDS PostgreSQL 6.4 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling PostgreSQL 6.4 or above on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click Here

Connection to Scriptcase

- **PostgreSQL 6.4 or above:** Click Here

Enabling Amazon RDS PostgreSQL 6.3 or below

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling PostgreSQL 6.3 or below on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click Here

Connection to Scriptcase

- **PostgreSQL 6.3 or below:** Click Here

Connection to Amazon RDS PostgreSQL

If you do not have the **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **PostgreSQL PDO:** Click Here
- To enable the **PostgreSQL 7 or above:** Click Here
- To enable the **PostgreSQL 6.4 or above:** Click Here
- To enable the **PostgreSQL 6.3 or below:** Click Here

ATTENTION: If you don't have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Driver enabled (**PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.**) and the Amazon RDS PostgreSQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS Postgres** database.

Connection

Enter the parameters for connecting to your Amazon RDS PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PostgreSQL PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-postgres.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **5432**.
- **Eschema**: Enter the schema for connection to the database.
- **Database NAme**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.

- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate):

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key):

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate):

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.

- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS MSSQL Server SRV

At Scriptcase, we have the following drivers available for connection to Azure MSSQL Server: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC .** If you are using your own pre-configured environment, the MSSQL Server extensions must be enabled manually in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 11 For SQL Server according to your PHP architecture .**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Listed below are the necessary files for the drivers to be enabled.

Required files:

x64

- Microsoft ODBC Driver 11 For SQL Server **(x64)**: Click Here

x86

- Microsoft ODBC Driver 11 For SQL Server **(x86)**: Click Here

Other Files

- PHP extensions **(php_pdo_sqlsrv_73_nts_x64 and php_sqlsrv_73_nts_x64)**: Click Here

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling Amazon RDS MSSQL Server SRV on Windows

IMPORTANT: If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, requiring only the installation of the client. **Proceed to item 3.**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64**. See the example below:

```
extension=php_pdo_sqlsrv_73_nts_x64
extension=php_sqlsrv_73_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 11 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **MSSQL Server NATIVE SRV:** Click here

Enabling Amazon RDS MSSQL Server SRV PDO

At Scriptcase, we have the following drivers available for connection to Azure MSSQL Server: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC .** If you are using your own pre-configured environment, the MSSQL Server extensions must be enabled manually in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 11 For SQL Server according to your PHP architecture .**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Listed below are the necessary files for the drivers to be enabled.

Required files:

x64

- Microsoft ODBC Driver 11 For SQL Server **(x64)**: Click Here

x86

- Microsoft ODBC Driver 11 For SQL Server **(x86)**: Click Here

Other Files

- PHP extensions **(php_pdo_sqlsrv_73_nts_x64 and php_sqlsrv_73_nts_x64)**: Click Here

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling Amazon RDS MSSQL Server SRV PDO on Windows

IMPORTANT: If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, requiring only the installation of the client. **Proceed to item 3.**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_73_nts_x64** and **php_sqlsrv_73_nts_x64**. See the example below:

```
extension=php_pdo_sqlsrv_73_nts_x64
extension=php_sqlsrv_73_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 11 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **MSSQL Server NATIVE SRV PDO:** Click here

Connection with Amazon RDS MSSQL

If you do not have the **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MSSQL Server NATIVE SRV**: Click here
- To enable the **MSSQL Server NATIVE SRV PDO**: Click here

  ATTENTION: If you do not have an Amazon RDS MSSQL Server base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**MSSQL Server NATIVE SRV or MSSQL Server NATIVE SRV PDO**) and the Amazon RDS MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS MSSQL Server** database.

Connection

Enter the parameters for connecting to your Amazon RDS PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **MSSQL Server Native SRV PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-mssql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS PDO DBLIB

In Scriptcase installed on Linux, we have the following driver available for connection to MSSQL Server: **PDO DBLIB.** If you are using your own pre-configured environment, MSSQL Server extensions must be enabled manually in PHP.

Configuring and enabling Amazon RDS PDO DBLIB on Linux

> **IMPORTANT:** If you are using the Scriptcase automatic installer, **the extensions are already enabled in PHP**, all that is missing is the configuration of the freetds file. **The procedure below is only for manual installations of Scriptcase. To see how to configure freetds in the automatic installation, scroll down to the configuration section by clicking here**

1 - If you do not have the extension enabled you can install the extension with this command in the terminal:

```
sudo apt-get install php8.1-pdo-dblib
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

Install the basic packages using the terminal for this procedure:

Ubuntu\Debian	RHEL\CentOS
sudo apt-get install unixodbc unixodbc-dev gcc nano wget make	sudo yum install unixODBC unixODBC-devel gcc nano wget make

- Download and Unzip the freeTDS:

```
wget https://www.freetds.org/files/stable/freetds-0.95.95.tar.gz

tar -zxf freetds-0.95.95.tar.gz
```

- Access the freeTDS directory and run the following command to compile and install FreeTDS:

```
sudo ./configure --with-tdsver=7.4 --with-unixodbc=/usr --disable-libiconv --disable-static --disable-threadsafe --enable-msdblib --disable-sspi --with-gnu-ld --enable-sybase-compat && make && make install
```

- Edit the FreeTDS configuration file:

Ubuntu\Debian	RHEL\CentOS
sudo nano/usr/local/etc/freetds.conf	sudo vim /usr/local/freetds/freetds.conf

- Add your MSSQL Server connection informations, like this example:

  ```
  [MSSQLServer] host = 192.18.72.03
  port = 1433
  tds version = 7.2
  ```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Setting up freetds - automatic installation

1 - Edit the FreeTDS configuration file:

Automatic installation directory
sudo nano /opt/Scriptcase/v9-php8/components/drivers/freetds/etc/freetds.conf

2 - Add your MSSQL Server connection informations, like in the example:

```
[MSSQLServer] host = 192.18.72.03
port = 1433
```

tds version = 7.4

3 - Restart Apache service

```
sudo service apachesc9php81 restart
```

Connection to Scriptcase

- **PDO DBLIB:** Click here

Connection with Amazon RDS PDO DBLIB

If you do not have the **PDO DBLIB** driver enabled, check below our documentation on how to enable this driver in Scriptcase for connection.

- To enable the **PDO DBLIB**: Click here

  ATTENTION: If you do not have an Amazon RDS MSSQL Server base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**PDO DBLIB**) and the Amazon RDS MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS MSSQL Server** database.

Connection

Enter the parameters for connecting to your Amazon RDS PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PDO DBLIB** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-mssql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

- **Encrypt**

  This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

- **trustservercertificate**

  Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

  If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

- **trustStore**

  The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

  When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

- **trustStorePassword**

  The password used to verify the integrity of trustStore data.

  If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

- **hostnameInCertificate**

  The hostname to use to validate the SQL Server TLS/SSL certificate.

  This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

  Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS PDO DBLIB

In Scriptcase installed on macOs, we have the following driver available for connection to MSSQL Server: **PDO DBLIB .** If you are using your own pre-configured environment, MSSQL Server extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Configuring and enabling the DBLIB PDO on macOS

If you have installed PHP 8.1 through our documentation using homebrew, **PDO DBLIB is already enabled in PHP**, requiring only the connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: Click here

Connection to Scriptcase

- **PDO DBLIB:** Click here

Connection with Amazon RDS PDO DBLIB

If you do not have the **PDO DBLIB** driver enabled, check below our documentation on how to enable this driver in Scriptcase for connection.

- To enable the **PDO DBLIB**: Click here

  ATTENTION: If you do not have an Amazon RDS MSSQL Server base created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**PDO DBLIB**) and the Amazon RDS MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS MSSQL Server** database.

Connection

Enter the parameters for connecting to your Amazon RDS PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PDO DBLIB** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-mssql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **1433**.
- **Database Name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above .**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS Oracle 8.0.5 or Higher

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed Scriptcase using the installer, it is not necessary to download **Microsoft Visual C ++ 2017** . For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017 .**

Required files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click here
- Microsoft Visual C++ **2017(x64)**: Click here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click here
- Microsoft Visual C++ **2017(x86)**: Click here

> **IMPORTANT:** PHP 7.3 is compatible with **Oracle Instant Client 11 or higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click here

Compatibility Table

Driver	Oracle database version
Oracle 8.0.5 or Higher	
Instant Client	**Oracle database version**
19.6.0.0	

> **ATTENTION:** To use the Oracle 11g **_Release 1 (11.1)** bank, you must use the **11.1 or 11.2** instant client.

Configuring and enabling Amazon RDS Oracle 8.0.5 on Windows

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Extract the previously downloaded Instant Client Basic package to your computer's root.

EX: C:\instantclient_19_6

2 - Paste in Scriptcase's **PHP** folderC:\Program Files\NetMake\v9-php73\components\php the DLLs listed below:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add the full path to the instant client in the **Path** parameter of the Environment Variables. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced system settings**

- Click on the **Advanced**> **Environment Variables tab .**

- Under **System variables**, select the **Path** item and click **Edit**.

- Click on the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **Oracle 8.0.5 or Higher:** Click Here

Enabling Amazon RDS Oracle PDO

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed Scriptcase using the installer, it is not necessary to download **Microsoft Visual C ++ 2017** . For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017 .**

Required files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click here
- Microsoft Visual C++ **2017(x64)**: Click here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click here
- Microsoft Visual C++ **2017(x86)**: Click here

> **IMPORTANT:** PHP 7.3 is compatible with **Oracle Instant Client 11 or higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click here

Compatibility Table

Driver	Oracle database version
Oracle PDO	
Instant Client	**Oracle database version**
19.6.0.0	

> **ATTENTION:** To use the Oracle 11g **_Release 1 (11.1)** bank, you must use the **11.1 or 11.2** instant client.

Configuring and enabling Amazon RDS Oracle PDO on Windows

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Extract the previously downloaded Instant Client Basic package to your computer's root.

EX: C:\instantclient_19_6

2 - Paste in Scriptcase's **PHP** folderC:\Program Files\NetMake\v9-php73\components\php the DLLs listed below:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add the full path to the instant client in the **Path** parameter of the Environment Variables. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced system settings**

- Click on the **Advanced**> **Environment Variables tab .**

- Under **System variables**, select the **Path** item and click **Edit**.

- Click on the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **Oracle PDO:** Click Here

Enabling Amazon RDS Oracle ODBC

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed Scriptcase using the installer, it is not necessary to download **Microsoft Visual C ++ 2017** . For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017 .**

Required files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click here
- Microsoft Visual C++ **2017(x64)**: Click here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click here
- Microsoft Visual C++ **2017(x86)**: Click here

> **IMPORTANT:** PHP 7.3 is compatible with **Oracle Instant Client 11 or higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click here

Compatibility Table

Driver	Oracle database version
Oracle ODBC	
Instant Client	**Oracle database version**
19.6.0.0	

> **ATTENTION:** To use the Oracle 11g **_Release 1 (11.1)** bank, you must use the **11.1 or 11.2** instant client.

Configuring and enabling Amazon RDS Oracle ODBC on Windows

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Extract the previously downloaded Instant Client Basic package to your computer's root.

EX: C:\instantclient_19_6

2 - Paste in Scriptcase's **PHP** folderC:\Program Files\NetMake\v9-php73\components\php the DLLs listed below:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add the full path to the instant client in the **Path** parameter of the Environment Variables. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced system settings**

- Click on the **Advanced**> **Environment Variables tab .**

- Under **System variables**, select the **Path** item and click **Edit**.

- Click on the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **Oracle ODBC:** Click Here

Enabling Amazon RDS Oracle 8

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed Scriptcase using the installer, it is not necessary to download **Microsoft Visual C ++ 2017** . For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017 .**

Required files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click here
- Microsoft Visual C++ **2017(x64)**: Click here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click here
- Microsoft Visual C++ **2017(x86)**: Click here

> **IMPORTANT:** PHP 7.3 is compatible with **Oracle Instant Client 11 or higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click here

Compatibility Table

Driver	Oracle database version
Oracle 8	
Instant Client	**Oracle database version**
19.6.0.0	

> **ATTENTION:** To use the Oracle 11g **_Release 1 (11.1)** bank, you must use the **11.1 or 11.2** instant client.

Configuring and enabling Amazon RDS Oracle 8 on Windows

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Extract the previously downloaded Instant Client Basic package to your computer's root.

EX: C:\instantclient_19_6

2 - Paste in Scriptcase's **PHP** folderC:\Program Files\NetMake\v9-php73\components\php the DLLs listed below:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add the full path to the instant client in the **Path** parameter of the Environment Variables. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced system settings**

- Click on the **Advanced**> **Environment Variables tab .**

- Under **System variables**, select the **Path** item and click **Edit**.

- Click on the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **Oracle 8:** Click Here

Connection to Amazon RDS Oracle

If you do not have the **Oracle 8.0.5 or Higher, Oracle PDO, and Oracle 8.** drivers enabled, check our documentation below on how to enable both in Scriptcase for connection.

- To enable the **Oracle 8.0.5 or Higher**: Click here
- To enable the **Oracle PDO**: Click here
- To enable the **Oracle 8**: Click here

CAUTION: If you do not have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS Oracle** database.

Connection

Enter the parameters for connecting to your Amazon RDS Oracle database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle PDO** Driver.
- **Database Name**: Enter the endpoint of the server where the database is installed along with **Service Name** or **Instance**.
  - **EX:** database-oracle.us-east-2.rds.amazonaws.com/ORCL
- **Scheme**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending

on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to Amazon RDS Oracle ODBC

If you do not have the **ODBC.** driver enabled, check our documentation below on how to enable it in Scriptcase for connection.

- To enable the **Oracle ODBC**: Click here

  CAUTION: If you do not have an Amazon RDS database created, check how to create it by clicking here.

Creating a System ODBC Data Source

Check below the entire step by step to create an ODBC data source to connect to Scriptcase.

> **NOTE**: ODBC must be created on the same server where ScriptCase is installed.

1 - Access your **ODBC Data Source Manager** and select according to your architecture. We will select the **x64** version:

2 - When entering the ODBC Data Sources Administrator, select the **DSN System** tab and click on **Add** to create your connection to your Oracle Database.

3 - After that, it is necessary to select the Driver to connect to Oracle. Select Driver: **Oracle in instantclient_19_6**

4 - Now, enter the connection parameters as in the example below:

- **Data Source Name**: Define here the name of the ODBC object that will be used when connecting to Scriptcase.
- **Description**: Write a definition for the Data Source, what is the purpose for example.
- **TNS Service Name**: Enter the endpoint of the server where the database is installed along with **Service Name** or **Instance**.
  - **EX:** database-oracle.us-east-2.rds.amazonaws.com/ORCL
  - For more information about Oracle's **Service Name**, click here.
- **User ID**: Inform the user that will be used in the connection for authentication.

5 - Test the connection to check that all parameters have been filled in correctly.

You will be asked to enter the password for connection.

6 - Click **OK** to complete this process.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS Oracle** database.

Connection

Enter the parameters for connecting to your Amazon RDS Oracle database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle ODBC** Driver.
- **Database Name**: Inform the Data Source Name configured.
  - **EX:** oracle
- **Scheme**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.

- **By default, Scriptcase leaves this option empty.**
- **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS Oracle 8.0.5 or Higher

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

> **IMPORTANT:** If you are using the automatic installer, **the extensions are already configured in PHP**, ready to connect if you have enabled them in the installation process. If you have not enabled them, see below how to enable them.

Configuring Amazon RDS Oracle 8.0.5 or Higher on Linux (Automatic installation)

The automatic installation of Scriptcase already comes with the Oracle drivers pre-configured, with only the **instant_client** to be configured to make the connection. Follow the steps described below to enable connection drivers.

1 - Download the Oracle Instant Client **12.1.0.2** by clicking here.

2 - After downloading, extract the file and copy the folder **instantclient** into the directory: **/opt/Scriptcase/v9-php81/components/drivers/**

3 - Restart the Apache sudo service apachesc9php81 restart

Configuring Amazon RDS Oracle 8.0.5 or Higher on Linux (Manual installation)

Required Files:

- **oci8.so** extension: Click Here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

> **WARNING**: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Drive	Oracle Database Version						
Instant Client	Oracle database version						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

E.g.

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

NOTE: If you are using CentOS/RHEL, it is not necessary to use 'export' before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8.0.5 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.

- **By default, Scriptcase enables this option.**
- **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS Oracle PDO

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

> **IMPORTANT:** If you are using the automatic installer, **the extensions are already configured in PHP**, ready to connect if you have enabled them in the installation process. If you have not enabled them, see below how to enable them.

Configuring Amazon RDS Oracle PDO on Linux (Automatic installation)

The automatic installation of Scriptcase already comes with the Oracle drivers pre-configured, with only the **instant_client** to be configured to make the connection. Follow the steps described below to enable connection drivers.

1 - Download the Oracle Instant Client **12.1.0.2** by clicking here.

2 - After downloading, extract the file and copy the folder **instantclient** into the directory: **/opt/Scriptcase/v9-php81/components/drivers/**

3 - Restart the Apache `sudo service apachesc9php81 restart`

Configuring Amazon RDS Oracle PDO on Linux (Manual installation)

Required Files:

- **oci8.so** extension: Click Here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

> **WARNING**: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Drive	Oracle Database Version						
Instant Client	Oracle database version						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

E.g.

NOTE: If you are using CentOS/RHEL, it is not necessary to use 'export' before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.

- **By default, Scriptcase enables this option.**
- **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS Oracle ODBC

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  CAUTION: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- **oci8.so** Extension: Click here

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

  IMPORTANT: PHP 7.3 is compatible with **Oracle Instant Client 11.2 or higher**.
  To download Oracle Instant Client lower than version **19.5.0.0**, it is necessary to have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 ou Higher**: Click here

Compatibility Table

Driver	Oracle database version
Oracle ODBC	
Instant Client	**Oracle database version**
12.1.0.2	

Configuring Amazon RDS Oracle ODBC on Linux

It is necessary to enable the Oracle extension in PHP and configure **instant_client** to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the **oci8.so** extension to the PHP extensions directory:

EX: sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and look for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/7.3/apache2/php.ini	/etc/php.ini

EX: extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/

E.g.

> **NOTE:** If you are using CentOS/RHEL, it is not necessary to use 'export' before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Connection to Scriptcase

- **Oracle ODBC:** Click here

Enabling Amazon RDS Oracle 8

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

> **IMPORTANT:** If you are using the automatic installer, **the extensions are already configured in PHP**, ready to connect if you have enabled them in the installation process. If you have not enabled them, see below how to enable them.

Configuring Amazon RDS Oracle 8 on Linux (Automatic installation)

The automatic installation of Scriptcase already comes with the Oracle drivers pre-configured, with only the **instant_client** to be configured to make the connection. Follow the steps described below to enable connection drivers.

1 - Download the Oracle Instant Client **12.1.0.2** by clicking here.

2 - After downloading, extract the file and copy the folder **instantclient** into the directory: **/opt/Scriptcase/v9-php81/components/drivers/**

3 - Restart the Apache

```
sudo service apachesc9php81 restart
```

Configuring Amazon RDS Oracle 8 on Linux (Manual installation)

Required Files:

- **oci8.so** extension: Click Here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

> **WARNING**: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle 8 Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Drive	Oracle Database Version						
Instant Client	Oracle database version						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

E.g.

NOTE: If you are using CentOS/RHEL, it is not necessary to use 'export' before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.

- **By default, Scriptcase enables this option.**
- **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to Amazon RDS Oracle

If you do not have the **Oracle 8.0.5 or Higher, Oracle PDO, and Oracle 8.** drivers enabled, check our documentation below on how to enable both in Scriptcase for connection.

- To enable the **Oracle 8.0.5 or Higher**: Click here
- To enable the **Oracle PDO**: Click here
- To enable the **Oracle 8**: Click here

CAUTION: If you do not have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS Oracle** database.

Connection

Enter the parameters for connecting to your Amazon RDS Oracle database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle PDO** Driver.
- **Database Name**: Enter the endpoint of the server where the database is installed along with **Service Name** or **Instance**.
  - **EX:** database-oracle.us-east-2.rds.amazonaws.com/ORCL
- **Scheme**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending

on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to Amazon RDS Oracle ODBC

If you do not have the **ODBC.** driver enabled, check our documentation below on how to enable it in Scriptcase for connection.

- To enable the **Oracle ODBC**: Click here

  CAUTION: If you do not have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS Oracle** database.

Connection

Enter the parameters for connecting to your Amazon RDS Oracle database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle ODBC** Driver.
- **Database Name**: Inform the Data Source Name configured.
  - **EX:** oracle
- **Scheme**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be

routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS Oracle 8.0.5 or Above

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Above, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Compatibility Table

Driver	Oracle database version
Oracle 8.0.5 or Higher	
Instant Client	**Oracle database version**
12.1.0.2	

Configuring Oracle 8.0.5 or Higher on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/7.3/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Connection to Scriptcase

- **Oracle 8.0.5 or Higher:** Click here

Enabling Amazon RDS Oracle PDO

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Above, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Compatibility Table

Driver	Oracle database version
Oracle PDO	
Instant Client	**Oracle database version**
12.1.0.2	

Configuring Oracle PDO on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/7.3/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Connection to Scriptcase

- **Oracle PDO:** Click here

Enabling Amazon RDS Oracle ODBC

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Above, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Compatibility Table

Driver	Oracle database version
Oracle ODBC	
Instant Client	**Oracle database version**
12.1.0.2	

Configuring Oracle ODBC on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/7.3/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Connection to Scriptcase

- **Oracle ODBC:** Click here

Enabling Amazon RDS Oracle 8

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Above, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Compatibility Table

Driver	Oracle database version
Oracle 8	
Instant Client	**Oracle database version**
12.1.0.2	

Configuring Oracle 8 on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/7.3/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Connection to Scriptcase

- **Oracle 8:** Click here

Connection to Amazon RDS Oracle

If you do not have the **Oracle 8.0.5 or Higher, Oracle PDO, and Oracle 8.** drivers enabled, check our documentation below on how to enable both in Scriptcase for connection.

- To enable the **Oracle 8.0.5 or Higher**: Click here
- To enable the **Oracle PDO**: Click here
- To enable the **Oracle 8**: Click here

CAUTION: If you do not have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS Oracle** database.

Connection

Enter the parameters for connecting to your Amazon RDS Oracle database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle PDO** Driver.
- **Database Name**: Enter the endpoint of the server where the database is installed along with **Service Name** or **Instance**.
  - **EX:** database-oracle.us-east-2.rds.amazonaws.com/ORCL
- **Scheme**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending

on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to Amazon RDS Oracle ODBC

If you do not have the **ODBC.** driver enabled, check our documentation below on how to enable it in Scriptcase for connection.

- To enable the **Oracle ODBC**: Click here

  CAUTION: If you do not have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS Oracle** database.

Connection

Enter the parameters for connecting to your Amazon RDS Oracle database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle ODBC** Driver.
- **Database Name**: Inform the Data Source Name configured.
  - **EX:** oracle
- **Scheme**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be

routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS MariaDB PDO

At Scriptcase, we have the following driver available for connection MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment MariaDB extension must be manually enabled in PHP.

Configuring and enabling Amazon MariaDB PDO on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

1 - In the php.ini file, located in C:\php, look for a line referring to MariaDB extension: **pdo_mysql** and remove the ; from the beginning of the line for the extension to be used by PHP. See the example below:

```
extension=pdo_mysql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php81 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **MariaDB PDO:** Click Here

Questions or Connection Problems?

Connection to Amazon RDS MariaDB

If you do not have the **MariaDB PDO** driver enabled, check below our documentation on how to enable in Scriptcase for connection.

- To enable the **MariaDB PDO**: Click Here

  Warning: If you don’t have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MariaDB PDO**) and the Amazon RDS MariaDB database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS MariaDB** database.

Connection

Enter the parameters for connecting to your Amazon RDS MariaDB database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MariaDB Driver for connection. In this example, we use the **MariaDB PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-amazon.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Amazon RDS MariaDB database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option .**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed

before the table names.
- **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS MariaDB PDO

At Scriptcase, we have the following driver available for connection to MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extensions must be manually enabled in PHP.

Configuring and enabling MariaDB PDO on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install MariaDB extension. See below for how to proceed.

```
sudo apt-get install php8.1-mysql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php sudo apt-get update
```

2 - Restart the Apache service through the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **MariaDB PDO:** Click Here

Connection to Amazon RDS MariaDB

If you do not have the **MariaDB PDO** driver enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MariaDB PDO**: Click Here

  CAUTION: If you don't have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MariaDB PDO**) and the Amazon RDS MariaDB database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS MariaDB** database.

Connection

Enter the parameters for connecting to your Amazon RDS MariaDB database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MariaDB Driver for connection. In this example, we use the **MariaDB PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-mariadb.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Amazon RDS MariaDB database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option .**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default .**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed

before the table names.
- **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Amazon RDS MariaDB PDO

At Scriptcase, we have the following driver available for connection to MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extension must be manually enabled in PHP.

Configuring and enabling MariaDB PDO on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have installed PHP 8.1 through our documentation using homebrew, **MariaDB is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 no macOS: Click Here

Connection to Scriptcase

- **MariaDB PDO:** Click Here

Questions or Connection Problems?

Connection to Amazon RDS MySQL

If you do not have the **MariaDB PDO** driver enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MariaDB PDO**: Click Here

  CAUTION: If you don't have an Amazon RDS database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MariaDB PDO**) and the Amazon RDS MariaDB database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Amazon RDS** connection.

4 - And choose the **Amazon RDS MariaDB** database.

Connection

Enter the parameters for connecting to your Amazon RDS MariaDB database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MariaDB Driver for connection. In this example, we use the **MariaDB PDO** Driver.
- **Server/Host (Name or IP)**: Enter the domain of the Amazon RDS server where the database is installed.
- **EX:** database-mysql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database name**: List and select the database you will connect to.
- **EX:** samples
- **Username**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Amazon RDS MariaDB database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option .**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether to display the tables for the informed owner.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **The utf8 charset is set by default.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed

before the table names.
- **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Google Cloud in Scriptcase

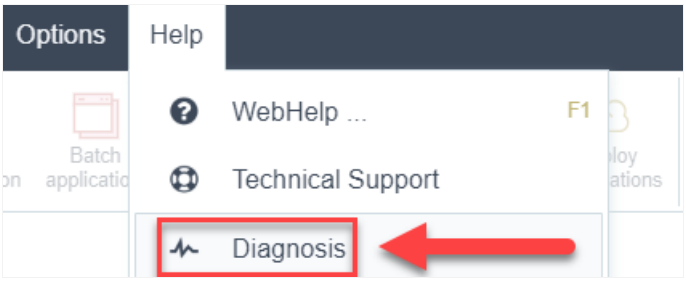
At Scriptcase, we have several databases (**MySQL, MariaDB, PostgreSQL and MSSQL Server**) and their respective drivers for connecting to Google Cloud. You must choose the most suitable for your environment and need.

If you do not have a Google Cloud database created from the ones available on Scriptcase, check below how to create a database:

- **Creation of Google Cloud MySQL database:** Click here
- **Creation of Google Cloud PostgreSQL database:** Click here
- **Creation of Google Cloud MSSQL Server database:** Click here

You can check your enabled drivers by accessing your diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.



- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `dominio.com/scriptcase/diagnosis.php`

Check below how to enable each database in Scriptcase according to your operating system:

MySQL

- **Windows:** Click here
- **Linux:** Click here
- **macOS:** Click here

MariaDB

- **Windows:** Click here
- **Linux:** Click here
- **macOS:** Click here

PostgreSQL

- **Windows:** Click here
- **Linux:** Click here
- **macOS:** Click here

MSSQL Server

- **Windows:** Click here
- **Linux:** Click here

- **macOS:** Click here

Enabling Google Cloud MySQLi

In Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling MySQLi on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to MySQL extensions: **mysqli** and **pdo_mysql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=mysqli
extension=pdo_mysql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **MySQLi:** ClicK Here

Enabling Google Cloud MySQL PDO

In Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling MySQLi on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to MySQL extensions: **mysqli** and **pdo_mysql** and remove the ; from the beginning of the lines for the extension to be used by PHP. See the example below:

```
extension=mysqli
extension=pdo_mysql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

Check below which Driver you want to connect with.

- **MySQL PDO:** Click here

Connection with Google Cloud MySQL

If you do not have the **MySQLi** or **MySQL PDO** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MySQLi**: Click Here
- To enable the **MySQL PDO**: Click Here

CAUTION: If you don’t have a Google Cloud database created, check how to create it by clicking [here][link_google]{: target = ‘_ blank’}.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MySQLi or MySQL PDO**) and the Google Cloud MySQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the database **Google Cloud MySQL**.

Connection

Enter the parameters for connecting to your Google Cloud MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MySQL Driver for connection. In this example, we use the **MySQL PDO** Driver.
- **SGDB server**: Enter the IP or domain of the Google Cloud server where the database is installed.
- **EX:** dominiodoserver.com ou 192.168.254.178
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Google Cloud MySQL database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Google Cloud MySQLi

In Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi .** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling Google Cloud MySQLi on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

1 - On your linux terminal, you only need to run one line to install MySQL extensions. See below for how to proceed.

```
sudo apt-get install php8.1-mysql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **MySQLi:** Click Here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud MySQL PDO

At Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling Google Cloud MySQL PDO on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install MySQL extensions. See below for how to proceed.

```
sudo apt-get install php8.1-mysql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **MySQL PDO:** Click Here

  Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Connection with Google Cloud MySQL

If you do not have the **MySQLi** or **MySQL PDO** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MySQLi**: Click Here
- To enable the **MySQL PDO**: Click Here

CAUTION: If you don't have a Google Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MySQLi or MySQL PDO**) and the Google Cloud MySQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the database **Google Cloud MySQL**.

Connection

Enter the parameters for connecting to your Google Cloud MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MySQL Driver for connection. In this example, we use the **MySQL PDO** Driver.
- **SGDB server**: Enter the IP or domain of the Google Cloud server where the database is installed.
- **EX:** dominiodoserver.com ou 192.168.254.178
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Google Cloud MySQL database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Google Cloud MySQLi

In Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling MySQLi on macOS

> If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP** , ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have PHP 8.1 installed through our documentation using homebrew, or __MySQL, PHP is not enabled, it is only necessary to connect. Check our documentation below for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 no macOS: Clique aqui

Connection to Scriptcase

- **MySQLi:** Clique aqui

Enabling Google Cloud MySQL PDO

In Scriptcase, we have the following drivers available for connection to MySQL: **MySQL PDO, MySQLi.** If you are using your own pre-configured environment, MySQL extensions must be manually enabled in PHP.

Configuring and enabling MySQLi on macOS

> If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP** , ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have PHP 8.1 installed through our documentation using homebrew, or __MySQL, PHP is not enabled, it is only necessary to connect. Check our documentation below for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 no macOS: Clique aqui

Connection to Scriptcase

- **MySQL PDO:** Clique aqui

Connection with Google Cloud MySQL

If you do not have the **MySQLi** or **MySQL PDO** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MySQLi**: Click Here
- To enable the **MySQL PDO**: Click Here

CAUTION: If you don’t have a Google Cloud database created, check how to create it by clicking [here][link_google]{: target = ‘_ blank’}.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MySQLi or MySQL PDO**) and the Google Cloud MySQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the database **Google Cloud MySQL**.

Connection

Enter the parameters for connecting to your Google Cloud MySQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MySQL Driver for connection. In this example, we use the **MySQL PDO** Driver.
- **SGDB server**: Enter the IP or domain of the Google Cloud server where the database is installed.
- **EX:** dominiodoserver.com ou 192.168.254.178
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Google Cloud MySQL database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Google Cloud PostgreSQL PDO

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL PDO on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines so that the extension is used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL 7 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL 7 or above on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C: \ php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines so that the extension is used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL 7 or above:** Click Here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL 6.4 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL 6.4 or above on Windows

> **IMPORTANT** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines so that the extension is used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL 6.4 or above:** Click Here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL 6.3 or below

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL 6.3 or below on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

1 - In the php.ini file, located in C:\php, look for lines referring to PostgreSQL extensions: **pgsql** and **pdo_pgsql** and remove the ; from the beginning of the lines so that the extension is used by PHP. See the example below:

```
extension=pgsql
extension=pdo_pgsql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **PostgreSQL 6.3 or below:** Click Here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Connection with Google Cloud PostgreSQL

If you do not have the **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable **PostgreSQL PDO:** Click Here
- To enable **PostgreSQL 7 ou acima:** Click Here
- To enable **PostgreSQL 6.4 ou acima:** Click Here
- To enable **PostgreSQL 6.3 ou abaixo:** Click Here

ATTENTION: If you don't have a Google Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.**) and the Google Cloud PostgreSQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the database **Google Cloud Postgres**.

Connection

Enter the parameters for connecting to your Google Cloud PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PostgreSQL PDO** Driver.
- **SGDB server**: Enter the IP or domain of the Google Cloud server where the database is installed.
- **EX:** domain.com ou 192.168.254.178
- **Port**: Enter the port for connection. By default, the defined port is **5432**.
- **Eschema**: Enter the schema for connection to the database.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the

parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate):

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key):

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate):

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed

- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection Questions or Problems?

Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL PDO

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL PDO on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL PDO:** Click Here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL 7 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL 7 or above on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL 7 or above:** click here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL 6.4 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL 6.4 or above on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL 6.4 or above:** click here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL 6.3 or below

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below .** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL 6.3 or below on Linux

> **IMPORTANT** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

1 - On your linux terminal, you only need to run one line to install PostgreSQL extensions. See below how to proceed.

```
sudo apt-get install php8.1-pgsql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **PostgreSQL 6.3 or below:** click here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Connection with Google Cloud PostgreSQL

If you do not have the **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable **PostgreSQL PDO:** Click Here
- To enable **PostgreSQL 7 ou acima:** Click Here
- To enable **PostgreSQL 6.4 ou acima:** Click Here
- To enable **PostgreSQL 6.3 ou abaixo:** Click Here

ATTENTION: If you don't have a Google Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.**) and the Google Cloud PostgreSQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the database **Google Cloud Postgres**.

Connection

Enter the parameters for connecting to your Google Cloud PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PostgreSQL PDO** Driver.
- **SGDB server**: Enter the IP or domain of the Google Cloud server where the database is installed.
- **EX:** domain.com ou 192.168.254.178
- **Port**: Enter the port for connection. By default, the defined port is **5432**.
- **Eschema**: Enter the schema for connection to the database.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the

parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate):

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key):

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate):

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed

- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection Questions or Problems?

Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL PDO

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configurando e habilitando o Google Cloud PostgreSQL PDO no macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: click here

Connection to Scriptcase

- **PostgreSQL PDO:** click here

> **Connection Questions or Problems?**
>
> Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL 7 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL 7 or above on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: click here

Connection to Scriptcase

- **PostgreSQL 7 or above:** click here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL 6.4 or above

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL 6.4 or above on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 on macOS: click here

Connection to Scriptcase

- **PostgreSQL 6.4 or above:** click here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Enabling Google Cloud PostgreSQL 6.3 or below

In Scriptcase, we have the following drivers available for connection to PostgreSQL: **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.** If you are using your own pre-configured environment, PostgreSQL extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PostgreSQL 6.3 or below on macOS

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase .**

If you have installed PHP 8.1 through our documentation using homebrew, **PostegreSQL is already enabled in PHP**, requiring only a connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 or macOS: click here

Connection to Scriptcase

- **PostgreSQL 6.3 or below:** click here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Connection with Google Cloud PostgreSQL

If you do not have the **PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable **PostgreSQL PDO:** Click Here
- To enable **PostgreSQL 7 ou acima:** Click Here
- To enable **PostgreSQL 6.4 ou acima:** Click Here
- To enable **PostgreSQL 6.3 ou abaixo:** Click Here

ATTENTION: If you don't have a Google Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**PostgreSQL PDO, PostgreSQL 7 or above, PostgreSQL 6.4 or above and PostgreSQL 6.3 or below.**) and the Google Cloud PostgreSQL database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the database **Google Cloud Postgres**.

Connection

Enter the parameters for connecting to your Google Cloud PostgreSQL database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the PostgreSQL Driver for connection. In this example, we use the **PostgreSQL PDO** Driver.
- **SGDB server**: Enter the IP or domain of the Google Cloud server where the database is installed.
- **EX:** domain.com ou 192.168.254.178
- **Port**: Enter the port for connection. By default, the defined port is **5432**.
- **Eschema**: Enter the schema for connection to the database.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the

parameters entered are correct.

Security

Use SSL

Allows you to define whether the connection will use SSL. Note that the option is disabled by default.

sslmode

sslmode	protection	MITM protection	Statement
disable	No	No	I don’t care about security, and I don’t want to pay the overhead of encryption.
allow	Maybe	No	I don’t care about security, but I will pay the overhead of encryption if the server insists on it.
prefer	Maybe	No	I don’t care about encryption, but I wish to pay the overhead of encryption if the server supports it.
require	Yes	No	I want my data to be encrypted, and I accept the overhead.
verify-ca	Yes	Depends on CA policy	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust.
verify-full	Yes	Yes	I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server I trust, and that it’s the one I specify.

sslrootcert (SSL Root Certificate):

This parameter refers to the SSL/TLS root certificate used to verify the authenticity of the certificate presented by the server during secure communication. Usage: The value of this parameter must be the full path to the file containing the root certificate.

sslkey (SSL Private Key):

This parameter represents the private key associated with the SSL/TLS certificate used by the server for authentication. Usage: The full path to the file containing the private key must be specified.

sslcert (SSL Certificate):

The sslcert parameter refers to the server’s SSL/TLS certificate, which is presented during the security negotiation process for authentication. Usage: The full path to the file containing the server’s SSL/TLS certificate must be provided.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed

- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection Questions or Problems?

Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Enabling Google Cloud SQL Server SRV

At Scriptcase, we have the following drivers available for connection to Google Cloud MSSQL Server: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC.** If you are using your own pre-configured environment, MSSQL Server extensions must be manually enabled in PHP. Google Cloud MSSQL Server SRV

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 13 For SQL Server according to your PHP architecture .**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

File list

Listed below are the necessary files for the drivers to be enabled.

- Download Microsoft ODBC Driver 13 For SQL Server **x64**
- Download Microsoft ODBC Driver 13 For SQL Server **x86**
- SQL Server extensions for PHP

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling Google Cloud SQL Server SRV on Windows

If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP** , requiring only the installation of the client. **Proceed to item 3**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64**. See the example below:

```
extension=pdo_sqlsrv_81_nts_x64
extension=sqlsrv_81_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 13 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click Next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart Apache service through **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php81 or Apache2.4 service and right click on that service, then click **Reiniciar**.

Connection to Scriptcase

- **MSSQL Server NATIVE SRV:** Click here

Enabling Google Cloud SQL Server SRV PDO

At Scriptcase, we have the following drivers available for connection to Google Cloud MSSQL Server: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC.** If you are using your own pre-configured environment, MSSQL Server extensions must be manually enabled in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 13 For SQL Server according to your PHP architecture .**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

File list

Listed below are the necessary files for the drivers to be enabled.

- Download Microsoft ODBC Driver 13 For SQL Server **x64**
- Download Microsoft ODBC Driver 13 For SQL Server **x86**
- SQL Server extensions for PHP

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling Google Cloud SQL Server SRV PDO on Windows

If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP** , requiring only the installation of the client. **Proceed to item 3**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64**. See the example below:

```
extension=pdo_sqlsrv_81_nts_x64
extension=sqlsrv_81_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 13 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click Next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart Apache service through **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php81 or Apache2.4 service and right click on that service, then click **Reiniciar**.

Connection to Scriptcase

- **MSSQL Server NATIVE SRV PDO:** Click here

Connection Questions or Problems?

Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Enabling Google Cloud SQL Server ODBC

At Scriptcase, we have the following drivers available for connection to Google Cloud MSSQL Server: **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC.** If you are using your own pre-configured environment, MSSQL Server extensions must be manually enabled in PHP.

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download Microsoft ODBC Driver 13 For SQL Server according to your PHP architecture .**

- **x86** Architecture = **32 bits**
- **x64** Architecture = **64 bits**

File list

Listed below are the necessary files for the drivers to be enabled.

- Download Microsoft ODBC Driver 13 For SQL Server **x64**
- Download Microsoft ODBC Driver 13 For SQL Server **x86**
- SQL Server extensions for PHP

Compatibility Table

	Server 2016	Server 2012 /R2	Windows 10	Windows 8 / 8.1	Server 2008 SP2 / R2 SP1	Windows 7 SP1	Windows Vista SP2
Native SRV PDO							
Native SRV							
ODBC							

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Configuring and enabling Google Cloud SQL Server ODBC on Windows

If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP** , requiring only the installation of the client. **Proceed to item 3**

1 - Extract the .zip file and copy the files from the **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64** .dll extensions previously downloaded to your PHP's ext folder.

2 - In the php.ini file, located in C:\php, add the lines referring to the MSSQL Server extensions **php_pdo_sqlsrv_81_nts_x64** and **php_sqlsrv_81_nts_x64**. See the example below:

```
extension=pdo_sqlsrv_81_nts_x64
extension=sqlsrv_81_nts_x64
```

3 - After downloading the Microsoft ODBC Driver 13 For SQL Server file, run the installer and follow the steps below:

4 - Accept the terms of the license agreement and click Next.

5 - Check the option **“ODBC Driver for SQL Server SDK”**, because the components that we will need are also in this option.

- **In this option, leave the option “Client Components” checked.**

6 - Click Install to start the client installation.

7 - The program features you selected are being installed.

8 - Click finish to exit.

9 - Restart Apache service through **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php81 or Apache2.4 service and right click on that service, then click **Reiniciar**.

Conexão com o Scriptcase

- **MSSQL Server ODBC:** click here

Connection Questions or Problems?

Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Connection with Google Cloud MSSQL

If you do not have the **MSSQL Server NATIVE SRV, MSSQL Server NATIVE SRV PDO and MSSQL Server ODBC** drivers enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MSSQL Server NATIVE SRV**: click here
- To enable the **MSSQL Server NATIVE SRV PDO**: click here

  ATTENTION: If you don't have a Google Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**MSSQL Server NATIVE SRV or MSSQL Server NATIVE SRV PDO**) and the Google Cloud MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the **Google Cloud SQL Server** database.

Connection

Enter the parameters for connecting to your Amazon RDS MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MSSQL Server Driver to connect. In this example, we use the **MSSQL Server NATIVE SRV PDO** Driver.
- **SGDB server**: Enter the domain of the Google Cloud server where the database is installed.
- **EX:** banco-mssql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port for connecting to the MSSQL Server. By default, the defined port is **1433**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

- **Encrypt**

  This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

- **trustservercertificate**

  Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

  If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

- **trustStore**

  The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

  When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

- **trustStorePassword**

  The password used to verify the integrity of trustStore data.

  If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

- **hostnameInCertificate**

  The hostname to use to validate the SQL Server TLS/SSL certificate.

  This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

  Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**

- **System Tables:** Selecting this option, the system tables of your database will be displayed.

- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**
- **Persistent Connection:** Define whether connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection Questions or Problems?

Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Connection to Amazon RDS MSSQL ODBC

If you do not have the **MSSQL Server ODBC** driver enabled, check below our documentation on how to enable this driver in Scriptcase for connection.

- To enable **MSSQL Server ODBC**: click here

  ATTENTION: If you don’t have a Google Cloud database created, check how to create it by clicking here.

Creating the DSN for connection

To connect to the **MSSQL Server ODBC** Driver, you need to configure a System DSN. Follow the steps described below to perform this configuration.

ATTENTION: ODBC must be created on the same Server where ScriptCase is installed.

1 - Access your **ODBC Data Source Manager** and select according to your architecture. We will select the x64 version:

2 - When entering the ODBC Data Sources Administrator, select the **DSN System** tab and click on **Add** to create your connection to your Database.

3 - After that, it is necessary to select the Driver to connect to the MSSQL Server. Select the Driver: **ODBC Driver 11 for SQL Server.**

4 - Now, you need to define the name of the DSN, the description and the database connection server.

- **Name:** Enter the name that the DSN will have so that you can use it in Scriptcase.
- **Description:** Add a description for the DSN.
- **Server:** inform the server, port and database instance that you will connect to.

4.1 - After entering the data, click on the **Next** button to enter the user name and password required for connection.

- **Login ID:** Inform the user to authenticate with the database.
- **Password:** Enter the corresponding password to authenticate with the informed user.

4.2 - Now, click **Next** to select the default database.

To finish, click **Next** and finally, click **Finish**.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Enabled Driver (**MSSQL Server NATIVE SRV or MSSQL Server NATIVE SRV PDO**) and the Azure MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection.

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Google Cloud** connection.

4 - And choose the **Google Cloud SQL Server** database.

Connection

Enter the parameters for connecting to your Azure MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MSSQL Server Driver to connect. In this example, we use the **MSSQL Server ODBC** Driver.
- **SGDB server**: Enter the DNS configured in your environment.
- **EX:** mssql
- **Port**: Enter the port to connect to the MSSQL Server. By default, the port defined is **1433**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

- **Encrypt**

  This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

- **trustservercertificate**

  Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

  If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

- **trustStore**

  The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

  When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

- **trustStorePassword**

  The password used to verify the integrity of trustStore data.

  If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

- **hostnameInCertificate**

  The hostname to use to validate the SQL Server TLS/SSL certificate.

  This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

  Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**
- **Persistent Connection:** Define whether connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection Questions or Problems?

Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Enabling Google Cloud PDO DBLIB

In Scriptcase installed on Linux, we have the following driver available for connection to MSSQL Server: **PDO DBLIB.** If you are using your own pre-configured environment, MSSQL Server extensions must be enabled manually in PHP.

Configuring and enabling Google Cloud PDO DBLIB on Linux

> **IMPORTANT:** If you are using the Scriptcase automatic installer, **the extensions are already enabled in PHP**, all that is missing is the configuration of the freetds file. **The procedure below is only for manual installations of Scriptcase. To see how to configure freetds in the automatic installation, scroll down to the configuration section by clicking here**

1 - If you do not have the extension enabled you can install the extension with this command in the terminal:

```
sudo apt-get install php8.1-pdo-dblib
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

Install the basic packages using the terminal for this procedure:

Ubuntu\Debian	RHEL\CentOS
sudo apt-get install unixodbc unixodbc-dev gcc nano wget make	sudo yum install unixODBC unixODBC-devel gcc nano wget make

- Download and Unzip the freeTDS:

```
wget https://www.freetds.org/files/stable/freetds-0.95.95.tar.gz

tar -zxf freetds-0.95.95.tar.gz
```

- Access the freeTDS directory and run the following command to compile and install FreeTDS:

```
sudo ./configure --with-tdsver=7.4 --with-unixodbc=/usr --disable-libiconv --disable-static --disable-threadsafe --enable-msdblib --disable-sspi --with-gnu-ld --enable-sybase-compat && make && make install
```

- Edit the FreeTDS configuration file:

Ubuntu\Debian	RHEL\CentOS
sudo nano/usr/local/etc/freetds.conf	sudo vim /usr/local/freetds/freetds.conf

- Add your MSSQL Server connection informations, like this example:

  ```
  [MSSQLServer] host = 192.18.72.03
  port = 1433
  tds version = 7.2
  ```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Setting up freetds - automatic installation

1 - Edit the FreeTDS configuration file:

Automatic installation directory
sudo nano /opt/Scriptcase/v9-php8/components/drivers/freetds/etc/freetds.conf

2 - Add your MSSQL Server connection informations, like in the example:

```
[MSSQLServer] host = 192.18.72.03
port = 1433
```

tds version = 7.4

3 - Restart Apache service

sudo service apachesc9php81 restart

Connection to Scriptcase

- **PDO DBLIB:** click here

Connection Questions or Problems?

Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Connection with Google Cloud PDO DBLIB

If you do not have the **PDO DBLIB** driver enabled, check below our documentation on how to enable this driver in Scriptcase for connection.

- To enable **PDO DBLIB**: click here

  ATTENTION: If you don’t have a Google Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Driver enabled (**PDO DBLIB**) and the Google Cloud MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Google Cloud** connection.

4 - And choose the **Google Cloud SQL Server database**.

Connection

Enter the parameters for connecting to your Amazon RDS MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MSSQL Server Driver to connect. In this example, we use the Driver **PDO DBLIB**.
- **SGDB server**: Enter the domain of the Google Cloud server where the database is installed.
- **EX:** banco-mssql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port to connect to the MSSQL Server. By default, the defined port is **1433**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Security tab, where the connection encryption settings are defined.

- **Encrypt**

  This property specifies whether communication with the SQL server should be encrypted. To enable encryption, you must set this property to “true”. This ensures that data sent between the client and server is protected by encryption.

- **trustservercertificate**

  Set to “true” to specify that the driver does not validate the server’s TLS/SSL certificate.

  If “true”, the server’s TLS/SSL certificate is automatically trusted when the communication layer is encrypted using TLS.

- **trustStore**

  The path (including the file name) to the certificate’s trustStore file. The trustStore file contains the list of certificates that the client trusts.

  When this property is not specified or is set to null, the driver relies on the trust manager factory query rules to determine which certificate store to use.

- **trustStorePassword**

  The password used to verify the integrity of trustStore data.

  If the trustStore property is set but the trustStorePassword property is not set, the integrity of the trustStore is not checked.

- **hostnameInCertificate**

  The hostname to use to validate the SQL Server TLS/SSL certificate.

  This property allows you to specify the expected hostname in the SQL server certificate. This is useful to ensure that the connection is only made to the correct server and not to a malicious server that may be using an invalid certificate.

  Note: This property is used in combination with the **encrypt**/authentication properties and the **trustServerCertificate** property. This property affects certificate validation if the connection uses **TLS** encryption and **trustServerCertificate** is set to “false”. Ensure that the value passed to hostNameInCertificate matches the Common Name (CN) or DNS name in the Subject Alternative Name (SAN) in the server certificate for a **TLS** connection to be successful. For more information about encryption support, see Understanding encryption support.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**

- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection Questions or Problems?

> Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Enabling Google Cloud PDO DBLIB

In Scriptcase installed on macOS, we have the following driver available for connection to MSSQL Server: **PDO DBLIB.** If you are using your own pre-configured environment, MSSQL Server extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Configuring and enabling the DBLIB PDO on macOS

If you have installed PHP 8.1 through our documentation using homebrew, **PDO DBLIB is already enabled in PHP**, requiring only the connection. Check below our documentation for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 or macOS: click here

Connection to Scriptcase

- **PDO DBLIB:** click here

Connection Questions or Problems?

Contact our [support][link_support]{:target='_ blank'} in case of connection problems or questions regarding this database.

Connection with Google Cloud PDO DBLIB

If you do not have the **PDO DBLIB** driver enabled, check below our documentation on how to enable this driver in Scriptcase for connection.

- To enable the **PDO DBLIB**: click here

  ATENÇÃO: Se você não possuir uma base de dados no Google Cloud criada, verifique como realizar a criação clicando aqui.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the Driver enabled (**PDO DBLIB**) and the Google Cloud MSSQL Server database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database> New connection** menu.

After that, a screen will appear with all database conexoes.

3 - Select the **Google Cloud** connection.

4 - And choose the **Google Cloud SQL Server database**.

Connection

Enter the parameters for connecting to your Amazon RDS MSSQL Server database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MSSQL Server Driver to connect. In this example, we use the Driver **PDO DBLIB**.
- **SGDB server**: Enter the domain of the Google Cloud server where the database is installed.
- **EX:** banco-mssql.us-east-2.rds.amazonaws.com
- **Port**: Enter the port to connect to the MSSQL Server. By default, the defined port is **1433**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** Selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the point . is selected as a separator.**
- **Persistent Connection:** Define whether conexoes will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection Questions or Problems?

> Contact our [support][link_support]{:target=’_ blank’} in case of connection problems or questions regarding this database.

Enabling Google Cloud MariaDB PDO

In Scriptcase, we have the following driver available for connection to MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extension must be manually enabled in PHP.

Configuring and enabling MariaDB on Windows

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - In the php.ini file, located in C:\php, look for a line referring to MariaDB extension: **pdo_mysql** and remove the ; from the beginning of the line for the extension to be used by PHP. See the example below:

```
extension=pdo_mysql
```

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php81 or Apache2.4 service and right click on this service, then **Restart**.

Connection to Scriptcase

Check below which Driver you want to connect with.

- **MariaDB PDO:** Click here

Connection with Google Cloud MariaDB

If you do not have the **MariaDB PDO** driver enabled, check below our documentation on how to enable in Scriptcase for connection.

- To enable the **MariaDB PDO**: Click Here

  Warning: If you don’t have a Google Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MariaDB PDO**) and the Google Cloud MariaDB database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the database **Google Cloud MariaDB**.

Connection

Enter the parameters for connecting to your Google Cloud MariaDB database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MariaDB Driver for connection. In this example, we use the **MariaDB PDO** Driver.
- **SGDB server**: Enter the IP or domain of the Google Cloud server where the database is installed.
- **EX:** dominiodoserver.com ou 192.168.254.178
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Google Cloud MariaDB database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Google Cloud MariaDB PDO

At Scriptcase, we have the following driver available for connection to MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extension must be manually enabled in PHP.

Configuring and enabling Google Cloud MariaDB PDO on Linux

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

1 - On your linux terminal, you only need to run one line to install MariaDB extensions. See below for how to proceed.

```
sudo apt-get install php8.1-mysql
```

- If you are unable to execute the above command because you cannot find packages, install the repository below and update.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

2 - Restart the Apache service via the terminal.

```
sudo service apache2 restart
```

Connection to Scriptcase

- **MariaDB PDO:** Click Here

Connection with Google Cloud MariaDB

If you do not have the **MariaDB PDO** driver enabled, check below our documentation on how to enable both in Scriptcase for connection.

- To enable the **MariaDB PDO**: Click Here

  CAUTION: If you don't have a Google Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MariaDB PDO**) and the Google Cloud MariaDB database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the database **Google Cloud MariaDB**.

Connection

Enter the parameters for connecting to your Google Cloud MariaDB database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MariaDB Driver for connection. In this example, we use the **MariaDB PDO** Driver.
- **SGDB server**: Enter the IP or domain of the Google Cloud server where the database is installed.
- **EX:** dominiodoserver.com ou 192.168.254.178
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Google Cloud MariaDB database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Google Cloud MariaDB PDO

In Scriptcase, we have the following driver available for connection to MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extension must be manually enabled in PHP.

Configuring and enabling MariaDB on macOS

> If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP** , ready for connection. **The procedure below is only for manual installations of Scriptcase.**

If you have PHP 8.1 installed through our documentation using homebrew, or **MariaDB**, PHP is not enabled, it is only necessary to connect. Check our documentation below for installing PHP 8.1 with homebrew on macOS.

PHP 8.1 no macOS: Clique aqui

Connection to Scriptcase

- **MariaDB PDO:** Clique aqui

Connection with Google Cloud MariaDB

If you do not have the **MariaDB PDO** driver enabled, check below our documentation on how to enable in Scriptcase for connection.

- To enable the **MariaDB PDO**: Click Here

  Warning: If you don’t have a Google Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver (**MariaDB PDO**) and the Google Cloud MariaDB database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the connection **Google Cloud**.

4 - And choose the database **Google Cloud MariaDB**.

Connection

Enter the parameters for connecting to your Google Cloud MariaDB database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **SGDB Driver**: Select the MariaDB Driver for connection. In this example, we use the **MariaDB PDO** Driver.
- **SGDB server**: Enter the IP or domain of the Google Cloud server where the database is installed.
- **EX:** dominiodoserver.com ou 192.168.254.178
- **Port**: Enter the port for connection. By default, the defined port is **3306**.
- **Database**: List and select the database you will connect to.
- **EX:** samples
- **User**: Inform the user to authenticate the connection to your database.
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Security

Accessing this tab, you can configure all the security of your Google Cloud MariaDB database.

- **Use SSL:** Choose whether the SSL protocol will be active on the connection.
  - **By default, Scriptcase disables this option.**

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** Selecting this option, the procedures of your database will be displayed.

Filters

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **The user must be in capital letters as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **utf8**.
  - **By default, the utf8 charset is set.**
- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.

    - **By default, the point . is selected as a separator.**

- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
    - **By default, Scriptcase disables this option.**

- **Use the schema before the table name:** Define whether the database schema is displayed before the table names.
    - **By default, Scriptcase enables this option.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Oracle Cloud in Scriptcase

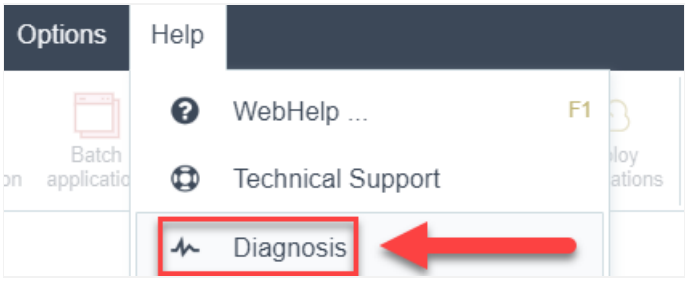
In Scriptcase, the **Oracle** database and its drivers are available for connection to Oracle Cloud. You must choose the most suitable driver for your environment and need.

If you do not have an Oracle Cloud database created from the ones available in Scriptcase, check below how to create a database:

- **Oracle Cloud database creation:** Click here

You can check your enabled drivers by accessing your diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.



- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `dominio.com/scriptcase/diagnosis.php`

Check below how to enable each database in Scriptcase according to your operating system:

Oracle

- **Windows:** Click here
- **Linux:** Click here
- **macOS:** Click here

Enabling Oracle Cloud 8.0.5 or Higher

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed Scriptcase using the installer, it is not necessary to download **Microsoft Visual C ++ 2017** . For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017 .**

Required files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click here
- Microsoft Visual C++ **2017(x64)**: Click here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click here
- Microsoft Visual C++ **2017(x86)**: Click here

> **IMPORTANT:** PHP 7.3 is compatible with **Oracle Instant Client 11 or higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click here

Compatibility Table

Driver	Oracle database version
Oracle 8.0.5 or Higher	
Instant Client	**Oracle database version**
19.6.0.0	

> **ATTENTION:** To use the Oracle 11g **_Release 1 (11.1)** bank, you must use the **11.1 or 11.2** instant client.

Configuring and enabling Oracle Cloud 8.0.5 on Windows

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Extract the previously downloaded Instant Client Basic package to your computer's root.

EX: C:\instantclient_19_6

2 - Paste in Scriptcase's **PHP** folderC:\Program Files\NetMake\v9-php73\components\php the DLLs listed below:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add the full path to the instant client in the **Path** parameter of the Environment Variables. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced system settings**

- Click on the **Advanced**> **Environment Variables tab .**

- Under **System variables**, select the **Path** item and click **Edit**.

- Click on the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **Oracle 8.0.5 or Higher:** Click Here

Enabling Oracle Cloud PDO

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed Scriptcase using the installer, it is not necessary to download **Microsoft Visual C ++ 2017** . For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017 .**

Required files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click here
- Microsoft Visual C++ **2017(x64)**: Click here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click here
- Microsoft Visual C++ **2017(x86)**: Click here

> **IMPORTANT:** PHP 7.3 is compatible with **Oracle Instant Client 11 or higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click here

Compatibility Table

Driver	Oracle database version
Oracle PDO	
Instant Client	**Oracle database version**
19.6.0.0	

> **ATTENTION:** To use the Oracle 11g **_Release 1 (11.1)** bank, you must use the **11.1 or 11.2** instant client.

Configuring and enabling Oracle Cloud PDO on Windows

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Extract the previously downloaded Instant Client Basic package to your computer's root.

EX: C:\instantclient_19_6

2 - Paste in Scriptcase's **PHP** folderC:\Program Files\NetMake\v9-php73\components\php the DLLs listed below:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add the full path to the instant client in the **Path** parameter of the Environment Variables. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced system settings**

- Click on the **Advanced**> **Environment Variables tab .**

- Under **System variables**, select the **Path** item and click **Edit**.

- Click on the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **Oracle PDO:** Click Here

Enabling Oracle Cloud ODBC

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed Scriptcase using the installer, it is not necessary to download **Microsoft Visual C ++ 2017** . For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017 .**

Required files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click here
- Microsoft Visual C++ **2017(x64)**: Click here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click here
- Microsoft Visual C++ **2017(x86)**: Click here

> **IMPORTANT:** PHP 7.3 is compatible with **Oracle Instant Client 11 or higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click here

Compatibility Table

Driver	Oracle database version
Oracle ODBC	
Instant Client	**Oracle database version**
19.6.0.0	

> **ATTENTION:** To use the Oracle 11g **_Release 1 (11.1)** bank, you must use the **11.1 or 11.2** instant client.

Configuring and enabling Oracle Cloud ODBC on Windows

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Extract the previously downloaded Instant Client Basic package to your computer's root.

EX: C:\instantclient_19_6

2 - Paste in Scriptcase's **PHP** folderC:\Program Files\NetMake\v9-php73\components\php the DLLs listed below:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add the full path to the instant client in the **Path** parameter of the Environment Variables. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced system settings**

- Click on the **Advanced**> **Environment Variables tab .**

- Under **System variables**, select the **Path** item and click **Edit**.

- Click on the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **Oracle ODBC:** Click Here

Enabling Oracle Cloud 8

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  NOTE: If you are using Scriptcase's automatic installer, the PHP architecture will be the same as the one you downloaded. **Download the Oracle instant client according to your PHP architecture.**

- Architecture **x86** = **32 bits**
- Architecture **x64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled. If you installed Scriptcase using the installer, it is not necessary to download **Microsoft Visual C ++ 2017** . For manual installations, **you must download and install both versions of Microsoft Visual C ++ 2017 .**

Required files:

x64

- Oracle Instant Client **19.6.0.0(x64)**: Click here
- Microsoft Visual C++ **2017(x64)**: Click here

x86

- Oracle Instant Client **19.6.0.0(x86)**: Click here
- Microsoft Visual C++ **2017(x86)**: Click here

> **IMPORTANT:** PHP 7.3 is compatible with **Oracle Instant Client 11 or higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **18.5.0.0 or Lower**: Click here

Compatibility Table

Driver	Oracle database version
Oracle 8	
Instant Client	**Oracle database version**
19.6.0.0	

> **ATTENTION:** To use the Oracle 11g **_Release 1 (11.1)** bank, you must use the **11.1 or 11.2** instant client.

Configuring and enabling Oracle Cloud 8 on Windows

The automatic installation of Scriptcase comes with pre-configured Oracle drivers, with only **instant_client** to be configured to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Extract the previously downloaded Instant Client Basic package to your computer's root.

EX: C:\instantclient_19_6

2 - Paste in Scriptcase's **PHP** folderC:\Program Files\NetMake\v9-php73\components\php the DLLs listed below:

- **oci.dll**
- **oraociei19.dll**
- **oraons.dll**

3 - Add the full path to the instant client in the **Path** parameter of the Environment Variables. Follow the steps below to perform this procedure:

- Go to Control Panel > System > **Advanced system settings**

- Click on the **Advanced**> **Environment Variables tab .**

- Under **System variables**, select the **Path** item and click **Edit**.

- Click on the **New** button and add the path to the instant client as shown below:

4 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Look for the ApacheScriptcase9php73 service and right click on this service, then **Restart**.

Connection to Scriptcase

- **Oracle 8:** Click Here

Connection to Oracle Cloud

If you do not have the **Oracle 8.0.5 or Higher, Oracle PDO, and Oracle 8.** drivers enabled, check our documentation below on how to enable both in Scriptcase for connection.

- To enable the **Oracle 8.0.5 or Higher**: Click here
- To enable the **Oracle PDO**: Click here
- To enable the **Oracle 8**: Click here

CAUTION: If you do not have an Oracle Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Oracle Cloud** connection.

Connection

Enter the parameters for connecting to your Oracle Cloud database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle PDO** Driver.
- **Database Name**: Enter the IP or domain of the server where the database is installed together with the **Service Name (Endpoint)** which is right after the IP in the image below. **EX:** 192.168.254.189/DB.oracle.com

Check which endpoint to use for the connection:

- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be

routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to Oracle Cloud ODBC

If you do not have the **Oracle ODBC.** driver enabled, check our documentation below on how to enable it in Scriptcase for connection.

- To enable the **Oracle ODBC**: Click here

  CAUTION: If you do not have an Oracle Cloud database created, check how to create it by clicking here.

Creating a System ODBC Data Source

Check below the entire step by step to create an ODBC data source to connect to Scriptcase.

> **NOTE**: ODBC must be created on the same server where ScriptCase is installed.

1 - Access your **ODBC Data Source Manager** and select according to your architecture. We will select the **x64** version:

2 - When entering the ODBC Data Sources Administrator, select the **DSN System** tab and click on **Add** to create your connection to your Oracle Database.

3 - After that, it is necessary to select the Driver to connect to Oracle. Select Driver: **Oracle in instantclient_19_6**

4 - Now, enter the connection parameters as in the example below:

- **Data Source Name**: Define here the name of the ODBC object that will be used when connecting to Scriptcase.
- **Description**: Write a definition for the Data Source, what is the purpose for example.
- **TNS Service Name**: It is the item is right after the IP on your Oracle Cloud console, as in the image below. **EX:** 192.168.254.189/DB.oracle.com

Check which endpoint to use for the connection:

- **User ID**: Inform the user that will be used in the connection for authentication.

5 - Test the connection to check that all parameters have been filled in correctly.

You will be asked to enter the password for connection.

6 - Click **OK** to complete this process.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Oracle Cloud** connection.

Connection

Enter the parameters for connecting to your Oracle Cloud database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle PDO** Driver.
- **Database Name**: Enter the Data Source Name created.
  - **EX:** oracle
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel.

This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Oracle Cloud 8.0.5 or Higher

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

> **IMPORTANT:** If you are using the automatic installer, **the extensions are already configured in PHP**, ready to connect if you have enabled them in the installation process. If you have not enabled them, see below how to enable them.

Configuring Oracle Cloud 8.0.5 or Higher on Linux (Automatic installation)

The automatic installation of Scriptcase already comes with the Oracle drivers pre-configured, with only the **instant_client** to be configured to make the connection. Follow the steps described below to enable connection drivers.

1 - Download the Oracle Instant Client **12.1.0.2** by clicking here.

2 - After downloading, extract the file and copy the folder **instantclient** into the directory: **/opt/Scriptcase/v9-php81/components/drivers/**

3 - Restart the Apache `sudo service apachesc9php81 restart`

Configuring Oracle Cloud 8.0.5 or Higher on Linux (Manual installation)

Required Files:

- **oci8.so** extension: Click Here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

> **WARNING**: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8.0.5	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Drive	Oracle Database Version						
Instant Client	Oracle database version						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

E.g.

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

NOTE: If you are using CentOS/RHEL, it is not necessary to use 'export' before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8.0.5 or Higher** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.

- **By default, Scriptcase enables this option.**
- **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Oracle Cloud PDO

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

> **IMPORTANT:** If you are using the automatic installer, **the extensions are already configured in PHP**, ready to connect if you have enabled them in the installation process. If you have not enabled them, see below how to enable them.

Configuring Oracle Cloud PDO on Linux (Automatic installation)

The automatic installation of Scriptcase already comes with the Oracle drivers pre-configured, with only the **instant_client** to be configured to make the connection. Follow the steps described below to enable connection drivers.

1 - Download the Oracle Instant Client **12.1.0.2** by clicking here.

2 - After downloading, extract the file and copy the folder **instantclient** into the directory: **/opt/Scriptcase/v9-php81/components/drivers/**

3 - Restart the Apache

```
sudo service apachesc9php81 restart
```

Configuring Oracle Cloud PDO on Linux (Manual installation)

Required Files:

- **oci8.so** extension: Click Here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

> **WARNING**: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Oracle database version**						

Drive	Oracle Database Version						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

E.g.

NOTE: If you are using CentOS/RHEL, it is not necessary to use 'export' before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the

parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**

- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Oracle Cloud ODBC

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Higher, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

> **IMPORTANT:** If you are using Scriptcase's automatic installer, **the extensions are already enabled in PHP**, ready for connection. **The procedure below is only for manual installations of Scriptcase.**

Prerequisites

Before proceeding with this documentation, check your PHP architecture. For drivers to be enabled correctly, **files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by the URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find the information regarding the architecture in phpinfo().

  CAUTION: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- **oci8.so** Extension: Click here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - ODBC Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - ODBC Package **12.1.0.2(i386/i686)**: Click here

> **IMPORTANT:** PHP 8.1 is compatible with **Oracle Instant Client 11.2 or higher**.
> To download Oracle Instant Client lower than version **19.5.0.0**, it is necessary to have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 ou Higher**: Click here

Compatibility Table

Drive	Oracle Database Version						
Oracle ODBC	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Drive	Oracle Database Version						
Instant Client	Oracle database version						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle Cloud ODBC on Linux

It is necessary to enable the Oracle extension in PHP and configure **instant_client** to make the connection. Follow the steps outlined below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the **oci8.so** extension to the PHP extensions directory:

EX: sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and look for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

EX: extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

WARNING: The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

E.g.

NOTE: If you are using CentOS/RHEL, it is not necessary to use ‘export’ before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Connection to Scriptcase

- **Oracle ODBC:** Click here

Enabling Oracle Cloud 8

In Scriptcase, we have the following drivers available for connections with Oracle: **Oracle PDO, Oracle 8.0.5 or Higher, Oracle ODBC and Oracle 8.** If you are using your own pre-configured environment, Oracle extensions must be manually enabled in PHP.

> **IMPORTANT:** If you are using the automatic installer, **the extensions are already configured in PHP**, ready to connect if you have enabled them in the installation process. If you have not enabled them, see below how to enable them.

Configuring Oracle Cloud 8.0.5 or Higher on Linux (Automatic installation)

The automatic installation of Scriptcase already comes with the Oracle drivers pre-configured, with only the **instant_client** to be configured to make the connection. Follow the steps described below to enable connection drivers.

1 - Download the Oracle Instant Client **12.1.0.2** by clicking here.

2 - After downloading, extract the file and copy the folder **instantclient** into the directory: **/opt/Scriptcase/v9-php81/components/drivers/**

3 - Restart the Apache sudo service apachesc9php81 restart

Configuring Oracle Cloud 8.0.5 or Higher on Linux (Manual installation)

Required Files:

- **oci8.so** extension: Click Here

Ubuntu x64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

Red-Hat/Centos/OpenSuse

x86_64

- Oracle Instant Client - Basic Package **12.1.0.2(x86_64)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(x86_64)**: Click here

i386/i686

- Oracle Instant Client - Basic Package **12.1.0.2(i386/i686)**: Click here
- Oracle Instant Client - Devel Package **12.1.0.2(i386/i686)**: Click here

> **WARNING**: PHP 8.1 is compatible with **Oracle Instant Client 11.2 or Higher**.
> To download Oracle Instant Client lower than version **19**, you must have an Oracle Account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Oracle PDO Driver Compatibility Table

Drive	Oracle Database Version						
Oracle 8	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Drive	Oracle Database Version						
Instant Client	Oracle database version						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - Install the packages below:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo apt-get update	sudo yum update
sudo apt-get install libaio1	sudo yum install libaio-devel
sudo apt-get install libncurses5	sudo yum install libaio
sudo apt-get install alien	sudo yum install glibc
sudo apt-get install gcc-multilib g++-multilib	sudo yum install compat-libstdc++-33
sudo apt-get install libpam0g	sudo yum install glibc-devel
sudo apt-get install unixodbc-dev unixodbc	sudo yum install libstdc++
	sudo yum install libstdc++
	sudo yum install pam
	sudo yum install ncurses-devel
	sudo yum install unixODBC

2 - Convert and install the previously downloaded **Instant Client Basic** packages on your Linux system:

UBUNTU\DEBIAN	CENTOS\RHEL
sudo alien oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
sudo alien oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm	sudo yum install oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
sudo dpkg -i oracle-instantclient12.1-basic_12.1.0.2.0-2_amd64.deb	
sudo dpkg -i oracle-instantclient12.1-devel_12.1.0.2.0-2_amd64.deb	

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

E.g. sudo cp oci8.so /usr/lib/php/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file and declare the Oracle extension at the end of the file.

UBUNTU\DEBIAN	CENTOS\RHEL
/etc/php/8.1/apache2/php.ini	/etc/php.ini

E.g. extension=oci8.so

5 - Edit the environment variable file and declare the variable responsible for identifying the Oracle library:

Enter the variable **LD_LIBRARY_PATH** and the path to the library in the file responsible for your system's environment variables. Check the path below according to your operating system:

> **WARNING:** The path to the library may depend on your installation, by default the path is this: /usr/lib/oracle/12.1/client64/lib/.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo nano /etc/apache2/envvars	sudo nano /etc/sysconfig/httpd

- Add this variable at the end of the file with the following syntax:

```
export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib/
```

E.g.

NOTE: If you are using CentOS/RHEL, it is not necessary to use 'export' before the variable.

6 - Restart the Scriptcase apache service for the changes to take effect. Use the commands below according to your operating system.

UBUNTU\DEBIAN	CENTOS\RHEL
sudo service apache2 restart	sudo systemctl restart httpd

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project, using the previously enabled Driver and the Oracle database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **Oracle** connection.

Connection

Enter the parameters for connecting to your Oracle database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver used to connect. In this example, we use the **Oracle 8** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed along with **Service Name**.
  - **E.g.** serverdomain.com/XE or 192.168.254.170/XE
  - For more information about Oracle's **Service Name**, click here.
- **Schema**: Inform the Scheme of specific tables created for the use of the informed user.
  - **This item is optional if you do not have a schema configured in your database.**
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
  - **By default, this option is left blank by Scriptcase.**
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.

- **By default, Scriptcase enables this option.**
- **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to Oracle Cloud

If you do not have the **Oracle 8.0.5 or Higher, Oracle PDO, and Oracle 8.** drivers enabled, check our documentation below on how to enable both in Scriptcase for connection.

- To enable the **Oracle 8.0.5 or Higher**: Click here
- To enable the **Oracle PDO**: Click here
- To enable the **Oracle 8**: Click here

CAUTION: If you do not have an Oracle Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Oracle Cloud** connection.

Connection

Enter the parameters for connecting to your Oracle Cloud database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle PDO** Driver.
- **Database Name**: Enter the server IP or domain where the database is installed along with **Service Name** or **Instance**.
  - **EX:** 192.168.254.189/DB.oracle.com
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to Oracle Cloud ODBC

If you do not have the **Oracle ODBC.** driver enabled, check our documentation below on how to enable it in Scriptcase for connection.

- To enable the **Oracle ODBC**: Click here

  CAUTION: If you do not have an Oracle Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Oracle Cloud** connection.

Connection

Enter the parameters for connecting to your Oracle Cloud database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle ODBC** Driver.
- **Database Name**: Enter the Data Source Name.
  - **EX:** oracle
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Enabling Oracle Cloud 8.0.5 or Higher

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Above, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Compatibility Table

Drive	Versão do banco de dados Oracle						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versão de base de dados Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8.0.5 or Higher on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/8.1/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Connection to Scriptcase

- **Oracle 8.0.5 or Higher:** Click here

Enabling Oracle Cloud PDO

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Above, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Compatibility Table

Drive	Versão do banco de dados Oracle						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versão de base de dados Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle PDO on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/8.1/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Connection to Scriptcase

- **Oracle PDO:** Click here

Enabling Oracle Cloud ODBC

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Above, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Compatibility Table

Drive	Versão do banco de dados Oracle						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versão de base de dados Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle ODBC on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

```
sudo cp oci8.so /usr/local/lib/php/pecl/20180731
```

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/8.1/php.ini) and declare the Oracle extension at the end of the file.

EX:

```
extension=oci8.so
```

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

```
sudo apachectl restart
```

Connection to Scriptcase

- **Oracle ODBC:** Click here

Enabling Oracle Cloud 8

At Scriptcase, we have the following drivers available for connection to Oracle: **Oracle 8.0.5 or Above, Oracle PDO, Oracle ODBC and Oracle 8 .** If you are using your own pre-configured environment, Oracle extensions must be enabled manually in PHP.

Prerequisites

Before proceeding at this documentation, check your PHP architecture. For drivers to be enabled correctly, **the files must be downloaded according to the architecture used.**

- In your Scriptcase, there is the file **info.php**, access it by URL: http://127.0.0.1:8092/scriptcase/info.php, where you will find information regarding architecture in the phpinfo().

  NOTE: Download the Oracle instant client according to your PHP architecture.

- Architecture **i386/i686** = **32 bits**
- Architecture **x86_64** = **64 bits**

Below are listed the files required for Oracle drivers to be enabled.

Required files:

- Extension **oci8.so**: Clique Aqui

x86_64

- Oracle Instant Client **12.1.0.2(x86_64)**: Clique Aqui

  IMPORTANT: The PHP 7.3 is compatible with **Oracle Instant Client 11.2 or Higher**.
  To download the Oracle Instant Client below the version **19.5.0.0**, it is necessary to have a Oracle account.

Other Files

- Oracle Instant Client **12.2.0 or Higher**: Click Here

Compatibility Table

Drive	Versão do banco de dados Oracle						
Oracle PDO	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c
Instant Client	**Versão de base de dados Oracle**						
19.6.0.0	9i	10g	11g R1	11g R2(11.2)	12c	18c	19c

Configuring Oracle 8 on MacOs

It is necessary to enable the Oracle extension in PHP and configure **Instant_Client** to make the connection. Follow the steps described below to enable the connection drivers.

1 - On your Mac terminal, log in as **root** and create a folder with instant client numbering in /usr/local/instantclient/12.1.0.2/.

EX:

- mkdir -p /usr/local/instantclient/12.1.0.2/

2 - Extract all content from the **Instant Client** downloaded previously on your Mac system and copy to the folder created in the previous step:

EX:

- unzip oci_client_macosx_12.zip
- cp -avr __MACOSX/ oci_client_macosx_12/ /usr/local/instantclient/12.1.0.2/

3 - Copy the previously downloaded **oci8.so** extension to the PHP extensions directory:

> **ATENÇÃO**: If you followed our documentation to install Scriptcase manually on macOs, the extension is already in the extensions directory. You can proceed to item **5**.

EX:

sudo cp oci8.so /usr/local/lib/php/pecl/20180731

Access the **info.php** file and search for the **extension_dir** line. It will tell you the exact path to the directory responsible for PHP extensions.

4 - Edit the **php.ini** file(/usr/local/etc/php/8.1/php.ini) and declare the Oracle extension at the end of the file.

EX:

extension=oci8.so

5 - Restart the apache service for the changes to be saved. Use the command below for this:

EX:

sudo apachectl restart

Connection to Scriptcase

- **Oracle 8:** Click here

Connection to Oracle Cloud

If you do not have the **Oracle 8.0.5 or Higher, Oracle PDO, and Oracle 8.** drivers enabled, check our documentation below on how to enable both in Scriptcase for connection.

- To enable the **Oracle 8.0.5 or Higher**: Click here
- To enable the **Oracle PDO**: Click here
- To enable the **Oracle 8**: Click here

CAUTION: If you do not have an Oracle Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Oracle Cloud** connection.

Connection

Enter the parameters for connecting to your Oracle Cloud database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle PDO** Driver.
- **Database Name**: Enter the server IP or domain where the database is installed along with **Service Name** or **Instance**.
  - **EX:** 192.168.254.189/DB.oracle.com
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connection to Oracle Cloud ODBC

If you do not have the **Oracle ODBC.** driver enabled, check our documentation below on how to enable it in Scriptcase for connection.

- To enable the **Oracle ODBC**: Click here

  CAUTION: If you do not have an Oracle Cloud database created, check how to create it by clicking here.

Creating a connection to Scriptcase

See below how to create a connection in your Scriptcase project, using the enabled Driver and the Oracle database.

1 - Access a project from your Scriptcase.

2 - Click the **New Connection** icon to create a connection

or access the **Database > New connection** menu.

After that, a screen will appear with all database connections.

3 - Select the **Oracle Cloud** connection.

Connection

Enter the parameters for connecting to your Oracle Cloud database as follows:

- **Connection name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the Oracle Driver to connect. In this example, we use the **Oracle ODBC** Driver.
- **Database Name**: Enter the Data Source Name.
  - **EX:** oracle
- **Username**: Inform the user to authenticate the connection to your Oracle database.
  - **For Oracle connections, the user must be capitalized, as in the example** .
- **Password**: Enter the password to complete the authentication process.
- **Test connection**: Click this button to get a response to the Scriptcase request to find out if the parameters entered are correct.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending on the owner or not.

Show

It allows the Oracle connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** Selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option .**
- **System Tables:** Selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX%% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **EX:**

- **Owner:** Inform the user who sees the tables informed for display.
  - **User must be capitalized as in the example above.**
- **View:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your Oracle database connection.

Advanced

In this tab, you have access to specific settings for the connection. The settings made in this session impact the data display and application performance.

- **Decimal Separator:** Select the separator type of the decimal records, between comma and period.
  - **By default, the . dot is selected as a separator.**
- **Persistent Connection:** Define whether connections will be terminated after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option .**
- **Use the schema before the table name:** Define whether the database schema will be displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **EX:**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

MariaDB with Scriptcase

In Scriptcase, we have one driver to connect with MariaDB. **If you are using your own pre-configured environment, MariaDB extensions must be manually enabled in PHP**.

You can check your enabled drivers by accessing your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your browser url:

`127.0.0.1:8092/scriptcase/diagnosis.php` or `domain.com/scriptcase/diagnosis.php`

If the desired driver is already enabled and you just want to know how to connect with Scriptcase, click on the link below according to the driver that will be used.

Environment	MariaDB PDO
Windows	See how to connect
Linux	See how to connect
macOS	See how to connect

Connecting with MariaDB PDO

In Scriptcase, we have the following driver available for connection with MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extension must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Windows, you'll need to enable the MariaDB extension on **php.ini** file. Check below how to do it.

1 - In the php.ini file, located in C:\php, uncomment the line referring to the MariaDB extension **php_pdo_mysql** removing the ; from the beginning of the line. See the example below:

2 - Restart the Apache service using the **Task Manager**.

- Open the Task Manager and click on the **Services** tab.
- Search for the Apache2.4 or ApacheScriptcase9php81 service and right click on this service, then **Restart**.

Creating a Connection in Scriptcase

See below how to create a connection in your Scriptcase project using the MariaDB database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **MariaDB** connection.

Connection

Enter the parameters for connecting to your MariaDB database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MariaDB Driver used to connect. In this example, we use the **MariaDB PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Username**: Inform the user to authenticate the connection to your MariaDB database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new MariaDB database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the

parameters entered are correct.

Security

On this tab you can set your SSL encryption for your database connection. Check below how to do this.

Use SSL

Activates or deactivates the use of secure connections with MariaDB.

Client Key

Path to the client’s private identification key in the format PEM.

Client Certificate

Path to the client’s public certificate key.

CA Path

Path to the directory that contains the Certificate of Authority (CA) in the format PEM, if used, it should specify the same certificate used by the server.

CA Certificate

Path to the Certificate of Authority (AC) in the format PEM. This option, if used, should specify the same certificate as the server.

Specific Chipper

A list of permitted numbers to use with the cryptography of the connection. If one of the numbers is not supported, the SSL connection won’t work.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the MariaDB connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with MariaDB PDO

In Scriptcase, we have the following driver available for connection with MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extension must be manually enabled in PHP.

Prerequisites

If you are using a manual installation on Linux, you'll need to install the MariaDB php extension. Check below how to do it.

1 - Access your **Linux** terminal and type this line below according to your OS to install MariaDB PDO driver.

Ubuntu	CentOS
sudo apt-get install php8.1-mysql	sudo yum install php-mysql

2 - Check if the MariaDB driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection with ScriptCase

See below how to create a connection in your Scriptcase project using the MariaDB database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **MariaDB** connection.

Connection

Enter the parameters for connecting to your MariaDB database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MariaDB Driver used to connect. In this example, we use the **MariaDB PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Username**: Inform the user to authenticate the connection to your MariaDB database.
- **Password**: Enter the password to complete the authentication process.

- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new MariaDB database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

On this tab you can set your SSL encryption for your database connection. Check below how to do this.

Use SSL

Activates or deactivates the use of secure connections with MariaDB.

Client Key

Path to the client's private identification key in the format PEM.

Client Certificate

Path to the client's public certificate key.

CA Path

Path to the directory that contains the Certificate of Authority (CA) in the format PEM, if used, it should specify the same certificate used by the server.

CA Certificate

Path to the Certificate of Authority (AC) in the format PEM. This option, if used, should specify the same certificate as the server.

Specific Chipper

A list of permitted numbers to use with the cryptography of the connection. If one of the numbers is not supported, the SSL connection won't work.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the MariaDB connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.
  - **By default, Scriptcase leaves this option empty.**
  - **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Connecting with MariaDB PDO

In Scriptcase, we have the following driver available for connections with MariaDB: **MariaDB PDO.** If you are using your own pre-configured environment, MariaDB extensions must be manually installed with PHP.

Prerequisites

If you are using a manual installation on MacOS, you'll need to install the PHP and the MariaDB driver will be enabled. Click here to see how to do it.

1 - Check if the MariaDB driver is enabled in your Scriptcase diagnosis. See below on how to locate your diagnosis and check if the driver is enabled.

- Accessing the top menu **Help > Diagnosis**, you can easily find it through the interface.

- Or, by accessing your Scriptcase installation path:

127.0.0.1/scriptcase/diagnosis.php or domain.com/scriptcase/diagnosis.php

Creating a Connection with ScriptCase

See below how to create a connection in your Scriptcase project using the MariaDB database.

1 - Access any project from your Scriptcase.

2 - Click on the **New Connection** icon to create a connection.

or access the **Database > New Connection** menu tab.

After that, a new page will appear with all database connections.

3 - Select the **MariaDB** connection.

Connection

Enter the parameters for connecting to your MariaDB database as follows:

- **Connection Name**: Define the name of your new connection. By default, Scriptcase adds the prefix **conn** along with the database name.
- **DBMS Driver**: Select the MariaDB Driver used to connect. In this example, we use the **MariaDB PDO** Driver.
- **Server/Host (Name or IP)**: Enter the IP or domain of the server where the database is installed.
  - **E.g.** serverdomain.com or 192.168.254.170
- **Port**: Inform the number port for your database server.
- **Username**: Inform the user to authenticate the connection to your MariaDB database.
- **Password**: Enter the password to complete the authentication process.
- **Database Name**: Type your database name to connect with him. Or, if you want to choose one different database, click on the button **List Database** to show all databases availabes for your user.
  - **Create Database:** Clicking on this button, you will create a new MariaDB database for use.
- **Test Connection**: Click on this button to get a response from the Scriptcase request to find out if the parameters entered are correct.

Security

On this tab you can set your SSL encryption for your database connection. Check below how to do this.

Use SSL

Activates or deactivates the use of secure connections with MariaDB.

Client Key

Path to the client’s private identification key in the format PEM.

Client Certificate

Path to the client’s public certificate key.

CA Path

Path to the directory that contains the Certificate of Authority (CA) in the format PEM, if used, it should specify the same certificate used by the server.

CA Certificate

Path to the Certificate of Authority (AC) in the format PEM. This option, if used, should specify the same certificate as the server.

Specific Chipper

A list of permitted numbers to use with the cryptography of the connection. If one of the numbers is not supported, the SSL connection won’t work.

Filter

Accessing this tab, you can configure which Database items will be displayed on the connection, depending or not on the owner.

Show

It allows the MariaDB connection to see tables, views, system tables and procedures depending on the items selected by the user. By default, the items **Tables** and **Views** are already selected by Scriptcase.

- **Tables:** By selecting this option, the tables in your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **Views:** By selecting this option, the views of your database will be displayed.
  - **By default, Scriptcase enables this option.**
- **System Tables:** By selecting this option, the system tables of your database will be displayed.
- **Procedures:** By selecting this option, the procedures of your database will be displayed.

Searches

Allows you to define which tables and owners are displayed.

- **Tables:** You can define in this option which tables will be displayed. The configuration can contain a PREFIX% or name of the tables to display.

- **By default, Scriptcase leaves this option empty.**
- **E.g.**

- **Owner:** Inform the user who sees the tables entered for display.
  - **The user must be in capital letters as in the example above.**
- **Show:** Choose whether tables for the informed owner are displayed.

  NOTE: By using table filtering, you eliminate unnecessary tables for your project and **improves the performance** of your database connection.

Advanced

In this tab, you have access to specific settings for the connection. The changes made in this session impact the data display and application performance.

- **client_encoding:** Select the encoding used in your database. In the example above, we use the client_encoding **UTF-8**.
- **Decimal Separator:** Select the type of separator for decimal records, between comma and period.
  - **By default, the period . is selected as a separator.**
- **Persistent Connection:** Define whether the connections will be closed after the execution of your scripts in Scriptcase applications.
  - **By default, Scriptcase disables this option.**
- **Use the schema before table name:** Define whether the database schema is displayed before the table names.
  - **By default, Scriptcase enables this option.**
  - **E.g.**

SSH

Here are the SSH options available in the image and their descriptions regarding database connection:

Use SSH

Enables or disables the SSH tunnel for database connection. When enabled, the database connection will be routed through the SSH server.

SSH Server

The IP address or hostname of the SSH server that will be used for tunneling. This is the machine that will act as a bridge to access the database securely.

SSH Port

The port used to connect to the SSH server.

SSH User

The username used to authenticate with the SSH server. This user must have permissions to establish an SSH connection.

Authentication method

You can choose two options: **password** or **certificate.** When you choose one of the options, a new field will appear below for the configuration.

Private cert. file

The private key file used for authentication if the SSH server requires key-based authentication instead of a password.

Local port for port forwarding

The local port on the client machine that will be used for forwarding database traffic through the SSH tunnel. This port acts as a bridge between the database client and the remote database.

Database server from SSH

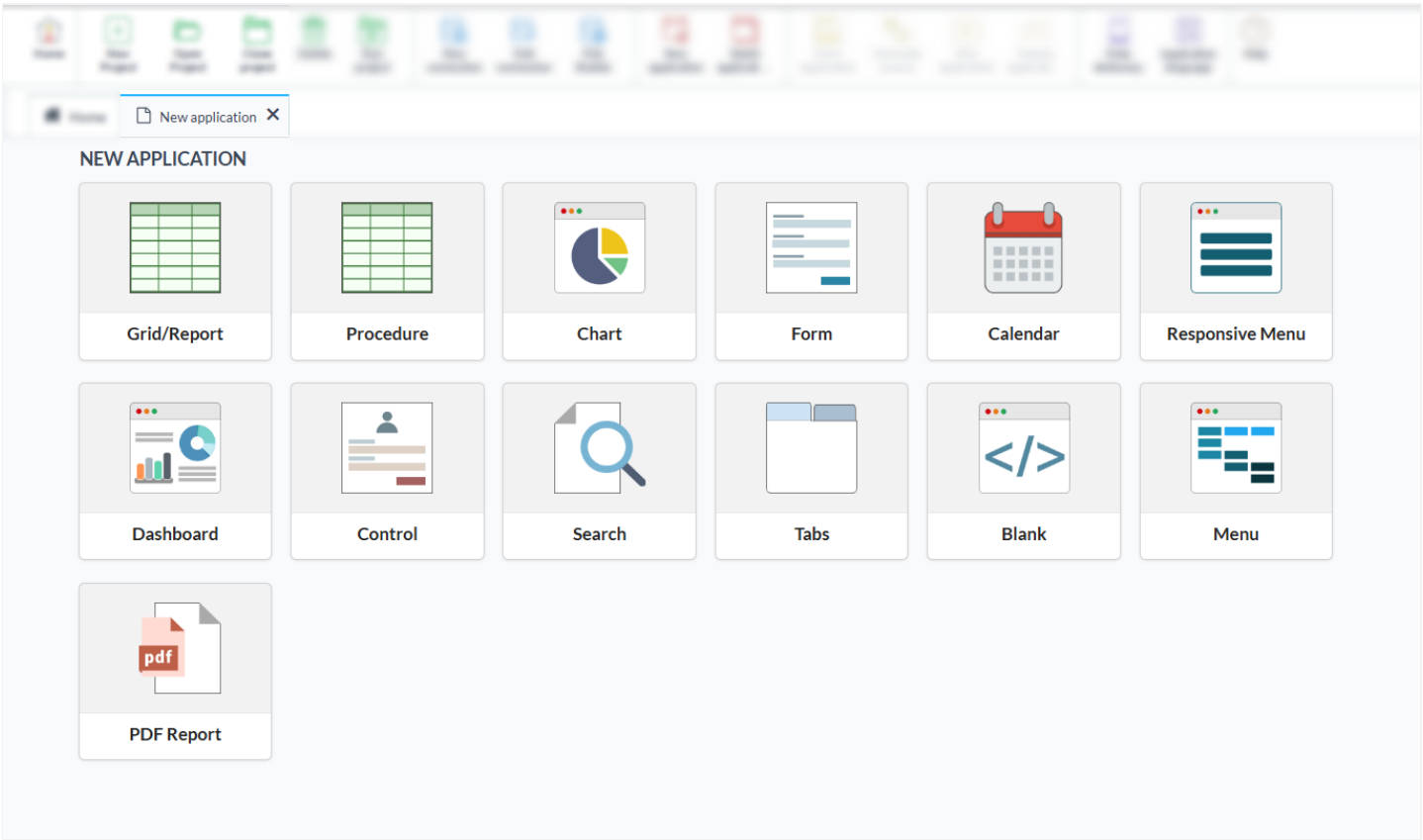
The hostname or IP address of the database server as seen from the SSH server. This is necessary when the database is only accessible within the SSH server’s network.

Database port from SSH

The port of the database server that will be accessed through the SSH tunnel. This should match the database service’s listening port.

Scriptcase Applications Overview

Scriptcase Applications



The Scriptcase offers a wide range of applications that allows the developer to create a complete system. The Scriptcase also allows the integration of the applications with external libraries, boosting its power in the systems development.

See below the available types of applications.

Grid

Application used to display data, this application serves as a report where we can do exports to PDF, XLS, XML, and other formats.

Procedure

Similar to the grid application, this application is available for few databases (MSSQL Server, Oracle and Db2) when selecting the connection, in case the user have a procedure that returns information it will be displayed in this special application.

Form

Application used to insert and update data, there are four different types of orientations.

Tabs

This application allows the creation of tabs where we can put up other applications like grids and forms.

Control

This application is used to create controls in a system, it is not linked to a table, and it needs the creation of one or more fields to its operation. A example of using is in the creation of a login screen.

Menu

It is used to create a hierarchical structure to navigate through the system's applications. There are two types of menu available in Scriptcase, Horizontal Menu and Tree Menu, that differs only in the preview of its items.

Responsive Menu

It is used to create a hierarchical structure to navigate through the system's applications.

Search

Application used to create search exclusively to forms.

Report PDF

Application used to generate reports using pre set formats, where we can position the data in the report.

Dashboard

Allows the developer to display applications inserted in widgets.

Blank

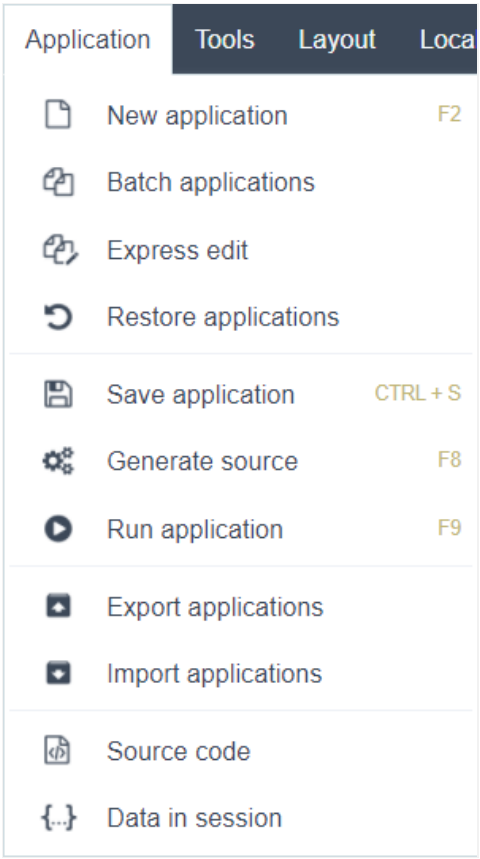
This application allows the insert of PHP code / HTML and displays the result of its processing. The advantage of using this application is the possibility of using macros and the native integrations with the other applications.

Calendar

Application where we can maintain an event schedule that can be synchronized to the google calendar.

Application Options Overview

The applications menu adds all features unique to applications like batch creation, express editing and application recovery.



Below is a brief explanation of each option.

New Application

Opens the application creation screen. Despite being extremely simple, each application has different options, to better untangle the creation of each of the applications, the articles were separated.

Below are the links to the documentation for creating each of the applications. **Click on the application you want to create**

- Grid
- Chart
- Form
- Control
- Filter
- Menu
- Tree Menu
- Calendar
- Tabs
- Report PDF
- Dashboard
- Blank
- Procedure

Create Batch application

Exclusive for creating Queries and Forms, it allows the creation of multiple applications with the editing link already configured.

Click here for more details.

Express Edition

Allows you to edit the project's applications simultaneously, allowing the definition of common values between them.

Click here for more details.

Restore Applications

This feature allows you to restore project applications that have undergone some recent change. Restore points are automatically set by the tool whenever a change is saved.

Click here for more details.

Save Applications

This option saves the application that is open in edit mode at the time of saving.

Click here for more details.

Generate Font

Generates the source code only for the application opened in editing mode.

Click here for more details.

Run Application

Saves and runs the application that is open in edit mode.

Click here for more details.

Export Applications

Export one or more applications from the project. In this type of export, all the files necessary for the selected applications to work are sent.

Click here for more details.

Import Applications

Allows you to import applications generated from the **Export Applications** option.

It is necessary that the export of the applications has been carried out in an equal or lower version for the project export to work. Thus, it is not possible to export an application from Scriptcase 9.7.002 and import an installation 9.7.001.

Click here for more details.

Source code

In this option it is possible to view the source code of the application that is open in editing mode.

For this feature to be available, the application's source code must have been previously generated.

Click here for more details.

Session data

In this option, you can see all the session variables available in the development environment, that is, it is the session tree stored by Scriptcase.

Click here for more details.

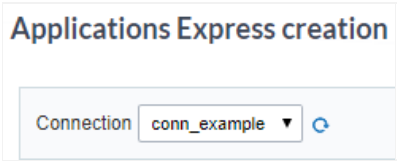
Batch Applications

Batch Applications

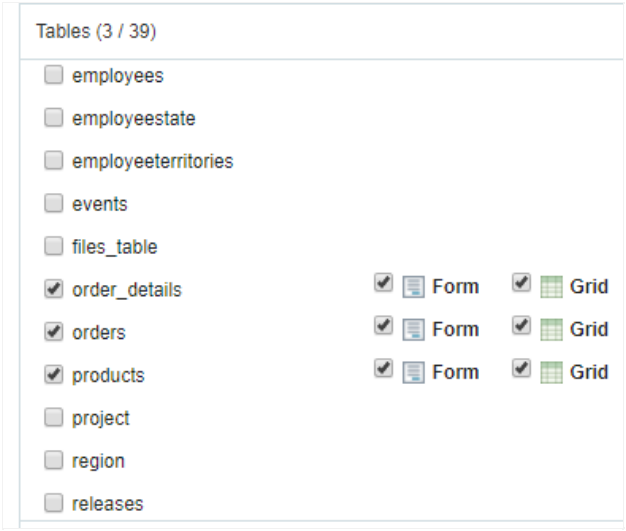
With this tool, it is possible to create multiple applications (Form and Grid).

When creating grids and forms using the same table, the applications are created with an application link between them, allowing to edit the record from the Grid application.

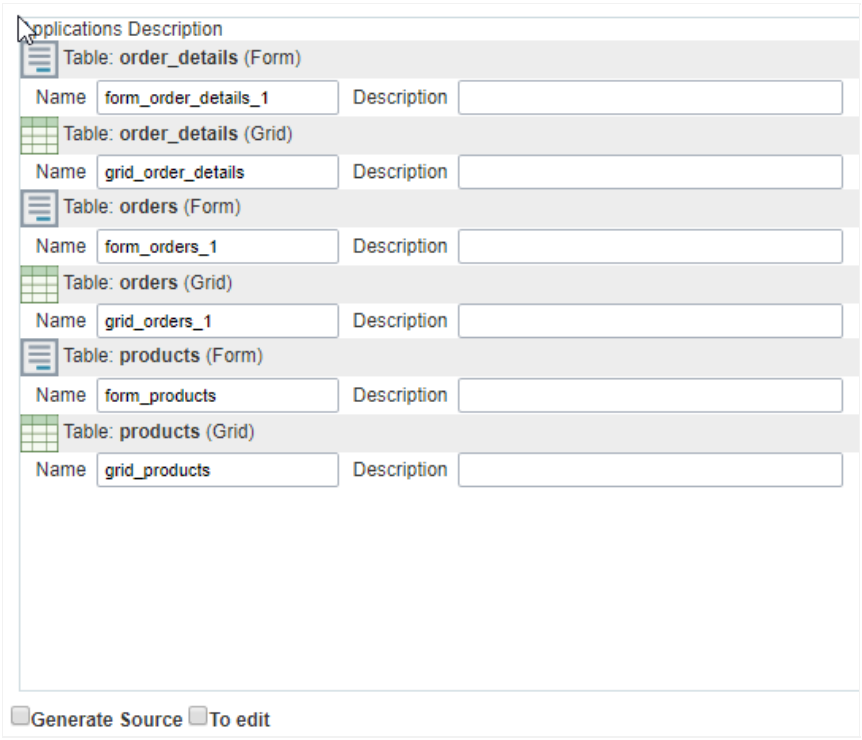
To start the process of creating the application, you need to select a connection so that the tables can be listed.



Next, you need to select the tables that are going to be used to create the applications. When selecting the tables, you can define which applications (Forms and Grids) are going to be created.



For last, you need to define the name, description and type, in case for the Forms.



- **Name** - Name of the application that is going to be created.
- **Description** - Application Description.
- **Type** - This option is only available for Forms, defines the type of form that is going to be created (Single Record, Multiple Rows, Editable Grid and Editable Grid “View”).
- **Generate Source** - Selecting this option all the applications to be created will have their source code generated.
- **To edit** - Selecting this option all the applications created will be open for modifications right after creation.

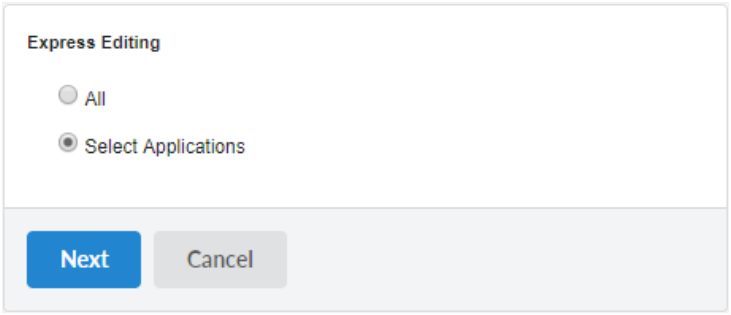
Express Edit

In this article, you learn how to edit various application at the same time. Using the Express Edit tool, you can define values for various common attributes for these applications. It is useful specially to add updates to your applications.

Only some settings are available within the Express Edit tool.

Select Application

Firstly, You need to inform which applications You want to edit.

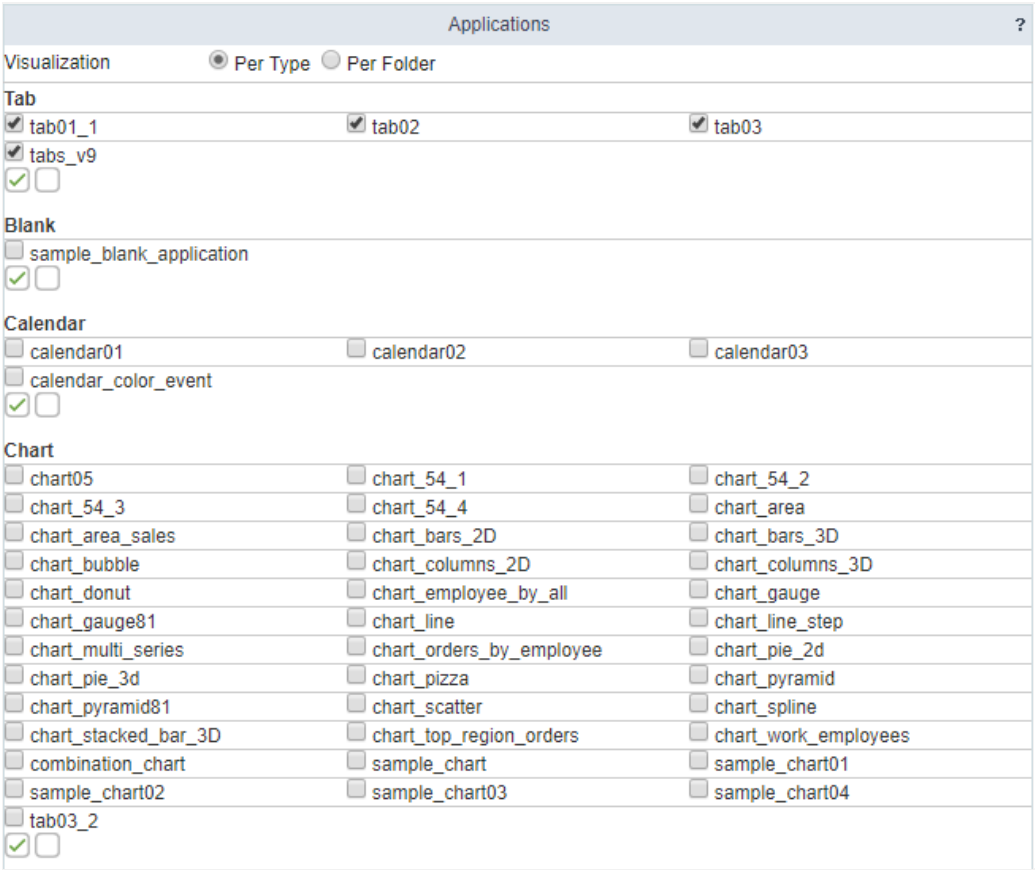


All

You edit all the applications in the project, with no exceptions.

Select Applications

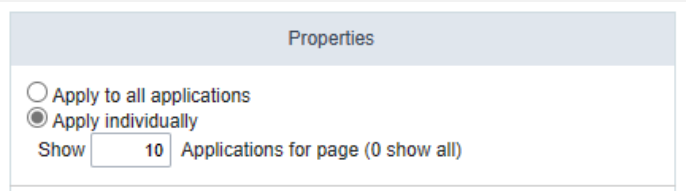
All the applications existing in the project are listed so that you can select the applications that you want to edit.



Properties

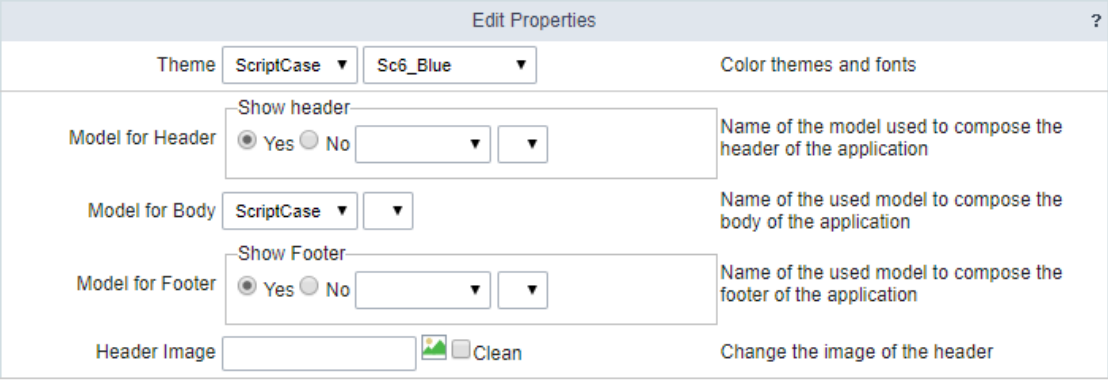
This page show all attributes enabled in the Express Edit, separated by categories.
These settings can be to apply to all applications simultaneously just select **Apply to all applications** or apply individually when selecting **Apply individually**.

If you use the **Apply individually** option, will be necessary to define to how many applications will be showed in the page. *Use 0 all application will be showed.*



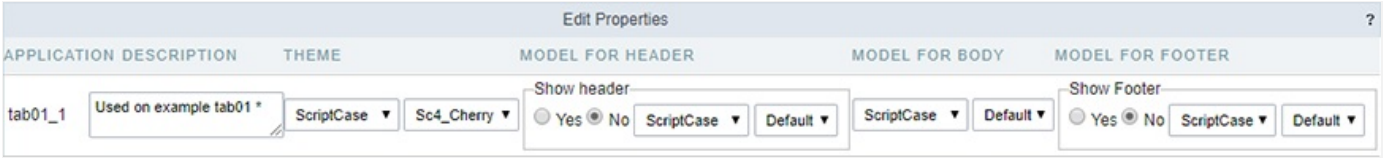
Apply to all applications

This option allows applying the same configurations, to all the selected applications, automatically.



Apply individually

This option allows you to apply the modifications individually for each application selected. This option contains not only the previous attributes, but also three exclusive attributes.



Categories - Layout/Theme

Header Image

Define the image for the header of the applications.

SweetAlert

It allows using the alerts messages with SweetAlert components.

SweetAlert position using Toast

This option define the position that Sweet Alert message using Toast will be shown.

Theme

It applies the desired Theme (Only the ones included in the Project properties) to all the applications.

Button

Allows you to define a Button Theme for selected applications

Template

Change the Header and Footer template of the applications.

Table Width

Defines the width of the table according to the unit of measure selected in the **Table Unit** option

Value is applied when selecting pixel or percentage

Table Unit

Sets the unit of measure for the Table Width option:

- Pixel - Fixed width of the application with its minimum value limited by the options available in the application.
- Percentage - adaptable width according to defined screen percentage. It has the limitation of application items.
- Automatic - Automatically fits the screen.

Fixed columns

Allows the choice of grid columns that will be fixed in the left corner of the screen if there is horizontal scrolling.

Fixed column hover

Display pinned column icon only on field label cell hover. Hover works only when the application is running as desktop, if it is running mobile the icon will always be displayed.

Fixed options' column

Pin the options column icons displayed in the row of the record regardless of having fixed columns selected.

Fixed group by

Allows the information of a grid's group by line to be fixed on the screen when scrolling horizontally.

Header Title

Define a title to the application.

This option will be enabled if you use the Apply individually option.

Toolbar buttons

Defines how the toolbar buttons will open. If it will open in modal or in a DIV below the toolbar.

Categories - SQL

Distinguish uppercase/loYourcase

It allows the Scriptcase to distinguish the letters uppercase and loYourcase.

Debug Mode

Define if the application is run will be with the debug mode enable. This option allows showing all SQL statements the application is executing.

Connection

It applies the desired connection (Connections created in the "New Connection" module) for the applications.

Connection for fields

It applies the desired connection (Connections created in the "New Connection" module) for the fields in the applications.

Categories - Mobile

Enable mobile optimization

Enables mobile optimization for adapting to mobile devices.

Enable scroll up button

It enables the displaying of the button back to top button in the Grid, Filter, Detail and Summary modules.

Scroll up button position

Defines the position of the back to the top button:

The options are:

- **Right** - Position the button in the bottom right corner;
- **Left** - Position the button in the bottom left corner;

Simplified toolbar

Defines the display format of the application's bottom toolbar, alternating between simplified when the option is enabled and traditional when these options are disabled.

To define the buttons that should compose your application's toolbar on mobile, access the Toolbar menu, then the Mobile tab.

Pin bottom toolbar

Defines the behavior of the lower toolbar when running the application on mobile devices with the Simplified toolbar option disabled.

When enabling these options, the lower toolbar will behave similarly to the upper toolbar, remaining fixed on the screen and thus facilitating access to the configured resources.

Categories - Security

Use Security

Define the use of security on the applications of the project.

Enable CSRF

It enables "Cross-Site Request Forgery" protection. (This option only applies for the Control and Form applications)

HttpOnly

Defines the use or not of the resource.

This option tells the browser that cookies should only be accessed via the HTTP protocol, preventing its access to scripting languages such as Javascript.

Session ID

Determines that the session ID must be stored in cookies on the client side and cannot be passed through URLs.

Cookie Secure

Determines that cookies should only be transmitted over secure connections.

Disable XSS Auditor

Determines whether or not to use the XSS Auditor option

Strict-Transport-Security

Determines that the application's pages can only be accessed through the HTTPS protocol.

X-Frame-Options

Determines the rule for including the application in iframes.

- SAMEORIGIN - allows only pages on the same site to include your application in iframes.
- DENY does - not allow your applications to be opened in iframes.

X-Content-Type-Options

Determines that the MIME types of the requested files must be respected by the browser.

Referrer-Policy

Determines how much reference information should be included in requests.

Feature-Policy

Determines which resources can be loaded by the browser. Add the rule on the same line being separated by semicolon(;)
Example: Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *;

Content-Security-Policy

Determines the behavior of certain browser APIs and functionality. Add the rule on the same line separated by semicolons(;)
Example: Content-Security-Policy: default-src https: 'unsafe-eval' 'unsafe-inline'; object-src 'none'

Permissions-policy

Determines the behavior of certain browser characteristics. Add the rule on the same line separated by a comma(,) **Example:** fullscreen=(), geolocation=()

Enable direct call by URL

Allow the applications to be called directly by the URL.

Export with Password

It allows defining a password in the Applications exportation.

Categories - Fields

Run content in JavaScript

Defines if the JavaScript code stored in the database runs on the browser.

Remove the HTML tags

Defines if the HTML code stored in the database will have the HTML tags removed for display.

Save HTML tags

Stores the HTML tags informed within the Application field to the database. (This option only applies for the Form application)

Fixed Label

It allows you to transform the Fields labels in static labels when you scroll down the page.

Use Select2

It allows you to apply the Select2 as default in the applications.

Display date format

Display the Field Format

Date display position

Position for the date format, on the right or bottom of the field.

Display message

Custom message for required fields.

Icon type

Choose whether the sort icons next to the field label will be the images defined in the application's theme or Font Awesome icons. This setting together with the selected icons for alphanumeric and numeric fields are used both in the grid and in the summary.

Alphanumeric fields

Choose the Font Awesome icon set to use for sorting alphanumeric fields.

Numeric fields

Choose the Font Awesome icon set to use when sorting numeric fields.

Text input in JavaScript

Stores the JavaScript informed within the Application field to the database. (This option only applies for the Form application)

Show HTML content

Defines if the HTML code stored in the database is interpreted or not by the browser. (This option only applies for the Grid application)

Categories - General

Documents Path

Absolute path where the application documents are stored.

Notify discarded changes

Notifies the user of changes that will be discarded when reloading the data.

Grid export modules

It allows you to select the Grid Modules in the exportation.

Description

You can define a description to the applications.

This option will be enabled if you use the Apply individually option.

Friendly URL

Define a friendly URL to application.

This option will be enabled if you use the Apply individually option.

Select columns of Export

It allows selecting the desired application columns in the exportation.

Display field error in a modal

Display the field error inside a message box opening in modal.

Field error position

Position where the field error will be displayed.

Timeout Error

Screen error message display time.

Highlight Results

Defines whether search results will be highlighted.

Image Directory

Base directory for the application images.

Language

It applies the selected language for all the selected applications in the project.

Hotkey Templates

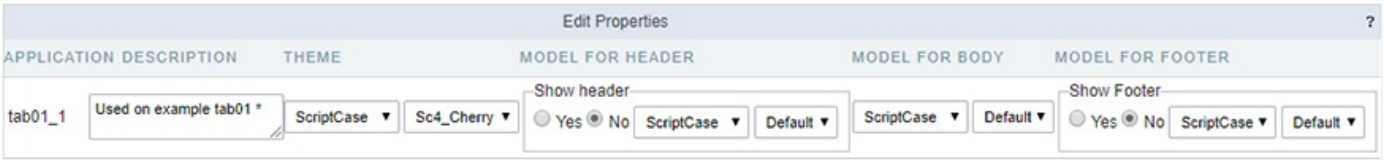
it allows configuring the hotkey template for all applications.

Image Viewer

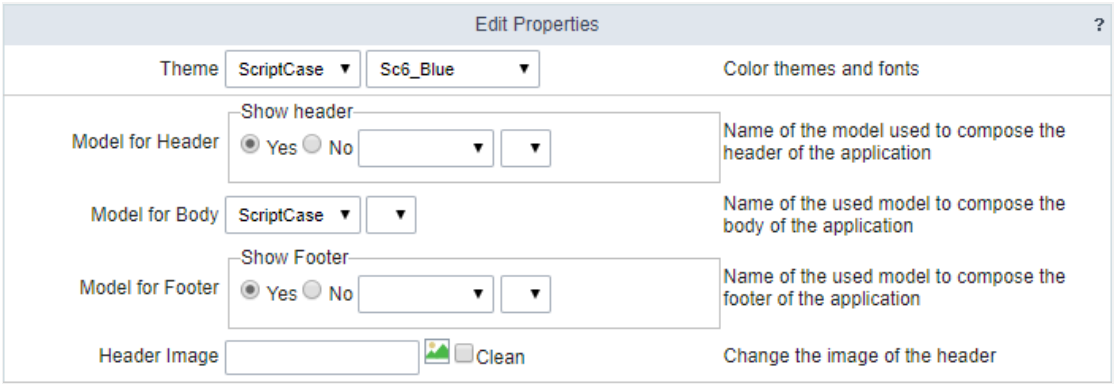
Defines whether the image viewer will be available.

Edit Properties

After selecting the modifications individually for all the select applications, It displays a list with all the selected attributes listed for modification.

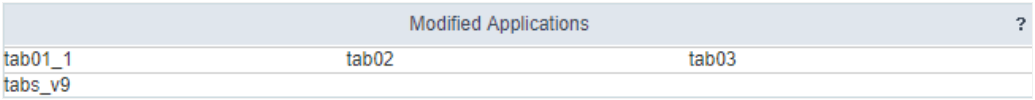


After selecting the modifications that you want set, you see a list with all the selected attributes.



Modified Applications

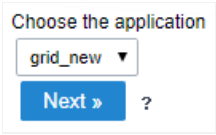
When you apply the modifications, It displays a list of the applications that Youre modified.



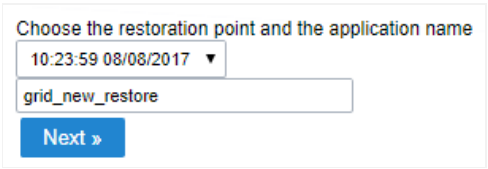
Restore Applications

Restore Applications

This feature allows to restore the project's applications that has been modified recently. When saving the application, ScriptCase stores automatically a limited number (defined in Settings > System Settings on the option Number of automatic application copies) of copies of the application, so that they can be restored. First, we need to select the application that you want to restore.



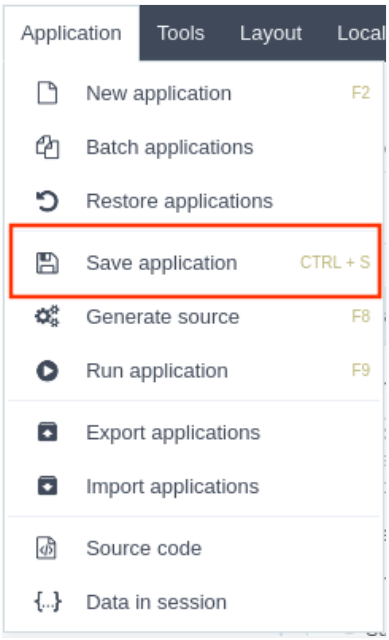
Previously, you need to select the restore point that you desire and also define a name for the restored application.



Save Applications

Save Applications

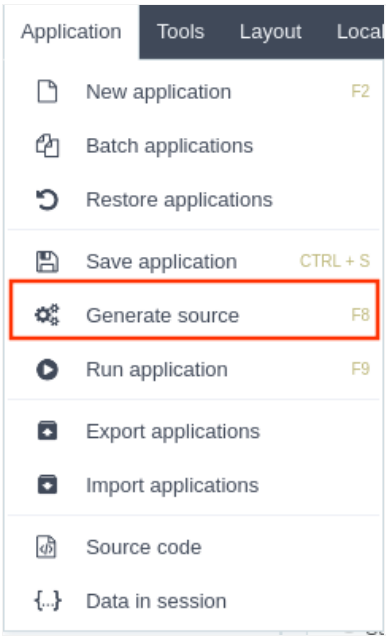
This option saves the application opened in the edition mode. This option can be found inside the applications menu or in the default toolbar, also being possible to use the shortcut CTRL+S.



Generate Source

Generate Source

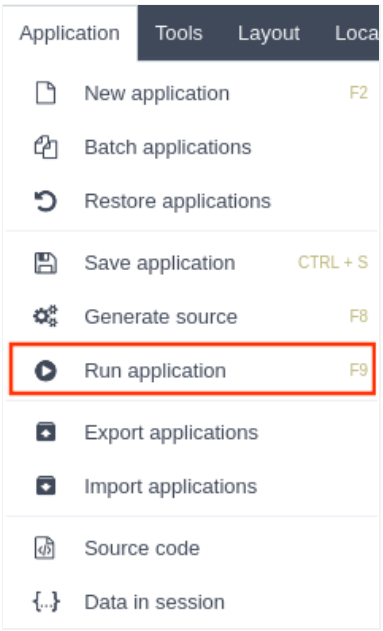
Different from the generate source code found in Project, this option saves and only generate the source code of the opened application in edition mode. This option can be found inside the menu Applications or in the default toolbar, and it is also possible to use the shortcut F8.



Execute Application

Execute Application

Saves and execute the application that is opened in the edition mode. This option can be found inside the Application menu of in the default toolbar, and it is also possible use the shortcut F9.



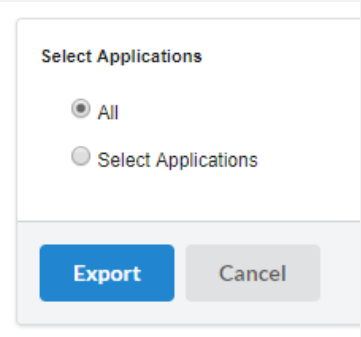
Export Applications

Export Applications

Every application of a project or some applications can be exported. In this export, not only the application but also the needed archives to its operation of the selected applications are exported.

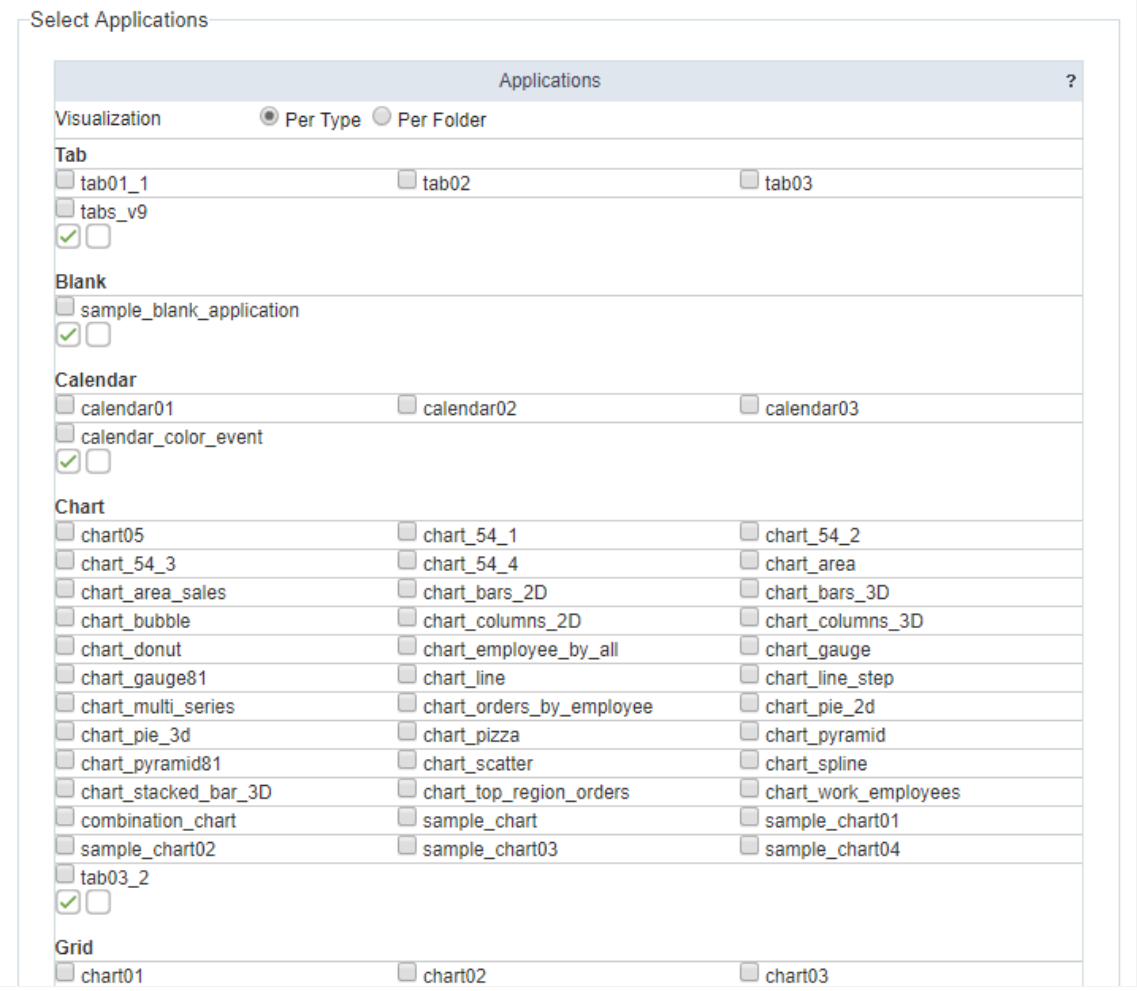
This option can be found inside the menu **Application**.

Firstly you need to select if you want to export all applications or if you want to select some applications.



When select all applications, it will be displayed with the export log and the link to donwload the generated file.

In this example, we choose to "**Select Applications**", in this case the applications are listed according to the type or folder.



To proceed, select the desired applications and click in export. After this it will be displayed a log of the exported files and a link

to download the generated file.

Backup Routine

Report
Time: 0:00:06,00
Items: 4337

MODULE	ITEMS	TIME
Buttons	19	0:00:00,04
Icons	4	0:00:00,01
Images	31	0:00:00,08
Theme	1	0:00:00,00
Footer Templates	2	0:00:00,00
Body Templates	33	0:00:00,05
Header Templates	2	0:00:00,01
PHP Libraries	1	0:00:00,01
Languages	50	0:00:00,13
Projects	1	0:00:00,00
Version	1	0:00:00,01
Connections	1	0:00:00,00
User	1	0:00:00,00
Application	337	0:00:01,88
Fields	3615	0:00:01,53
Events	237	0:00:00,60
Zip File	1	0:00:01,62

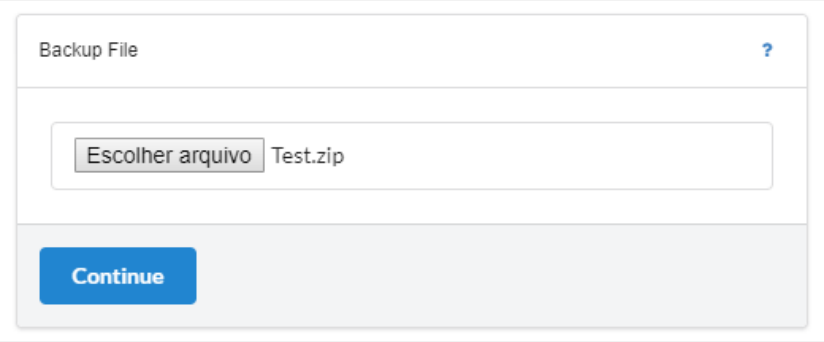
Backup file
sc9_201708081048_export_samples.zip

Import Applications

Import Applications

Allows the developer to import applications created in the same version of Scriptcase. This option can be found inside the menu **Application**.

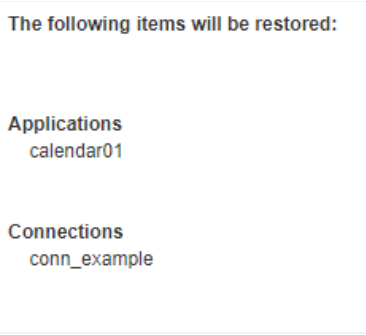
After the import application screen is displayed, select the file you want to import.



The options are:

Overwrite - Overwrites the project's files by the files that are being imported. **Don't overwrite** – Keeps the project's files, ignoring the files that are being imported. **Rename** – Keep the project's and import the files selected with the defined names.

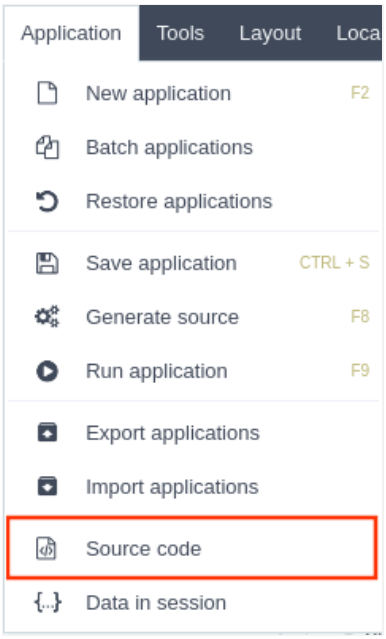
In the next step, show a summary that will be restored.



Source code

Source code

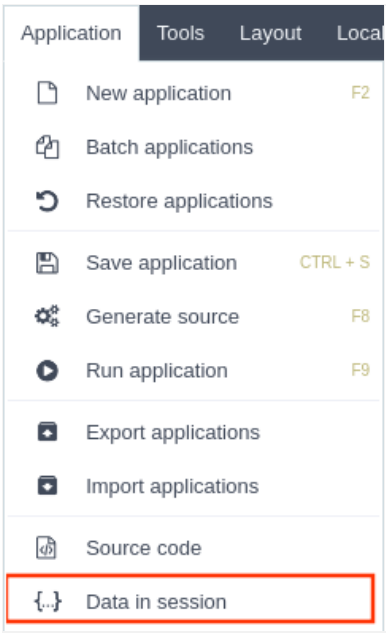
In this option is possible to see the application's source code after it generates.



Session data

Session data

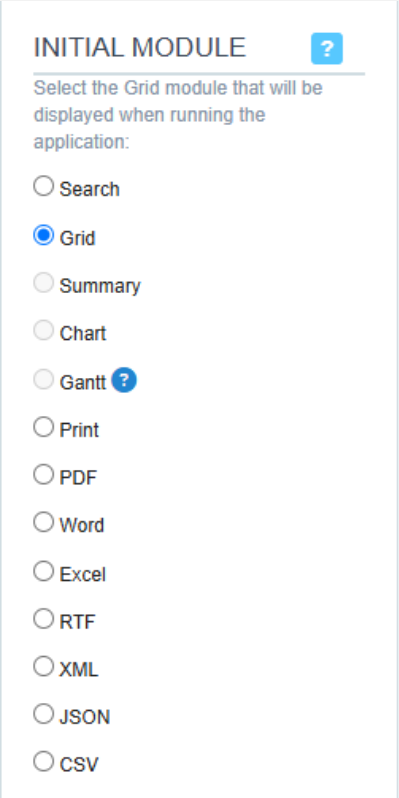
In this option, it is possible to see every session variable available in the development environment, it is the session tree that Scriptcase stores.



Grid Modules Settings

Initial Module

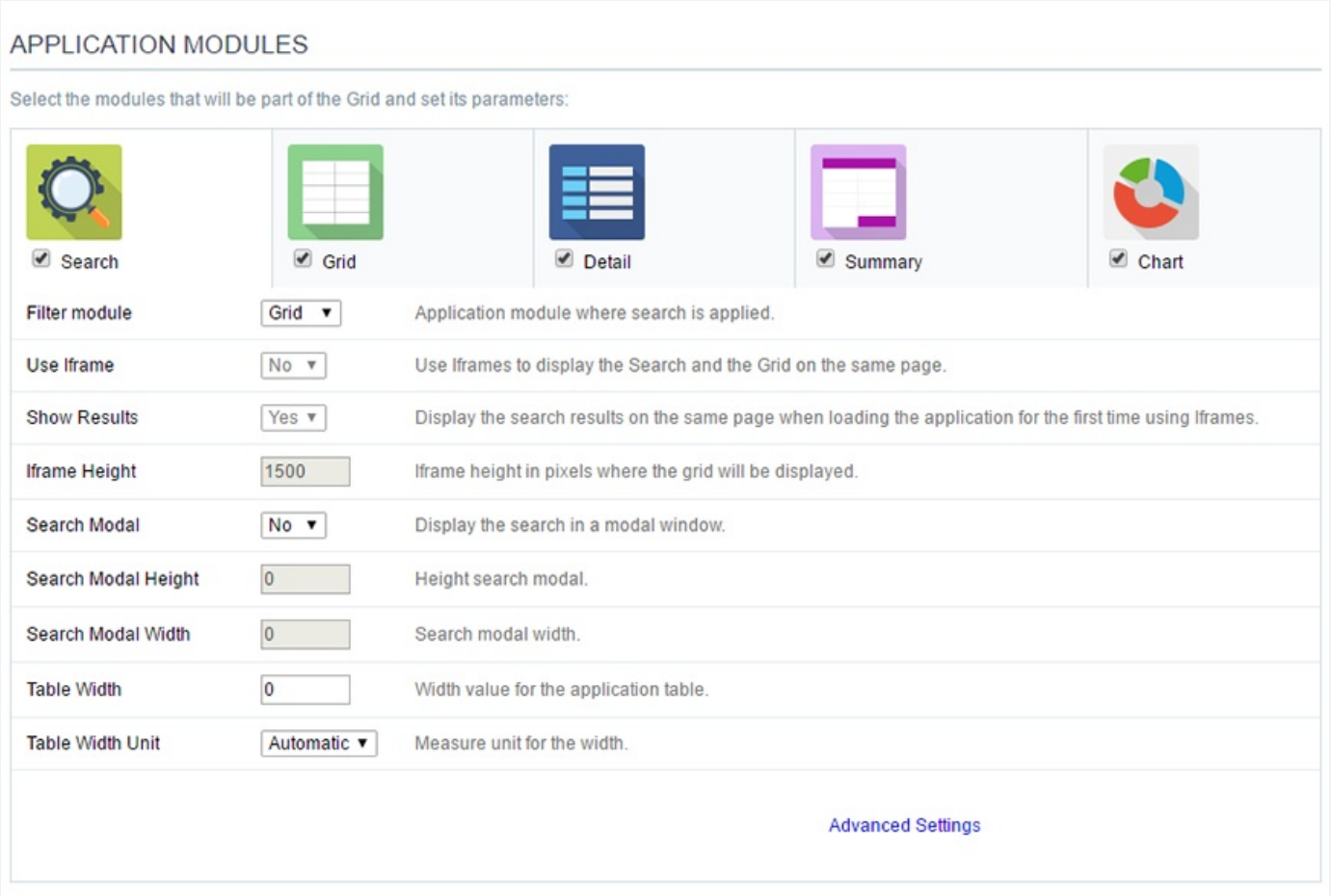
Allows you to set in which mode the application will start when executed.



Grid Application Modules

- **Search** : The application will start with the Advanced Search module, so you can filter the records before the next application, that can be a Grid, PDF, Summary, Chart, Print, PDF, Word, Excel, RTF, XML or CSV. You can configure these options within the Filter's settings.
- **Grid** : The application will start with the Grid module itself, this is the default option.
- **Summary** : The application will start with the Summary module. To use the Summary as initial application it's mandatory to create at least one Group, using the Static Group By settings.
- **Chart** : The application will start with the Chart module. To use the Chart as initial application it's mandatory to create at least one Group, using the Static Group By settings.
- **Gantt**: When this option is selected, the application will open directly in the **Gantt** view, displaying the schedule based on the Gantt configuration; to enable it, configure the necessary options and ensure the *Gantt* button is visible in the toolbar.
- **Print** : The application will start with the print mode, according to the select command.
- **PDF** : Application will be started with the option to view or download a PDF file.
- **Word** : Application will be started with the option to view or download a Word file(.doc or .docx).
- **Excel** : Application will be started with the option to view or download an Excel file (.xls or .xls).
- **RTF** : Application will be started with the option to view or download a RTF file (.tft).
- **XML** : Application will be started with the option to view or download a XML file.
- **CSV** : Application will be started with the option to view or download a CSV file.

Search



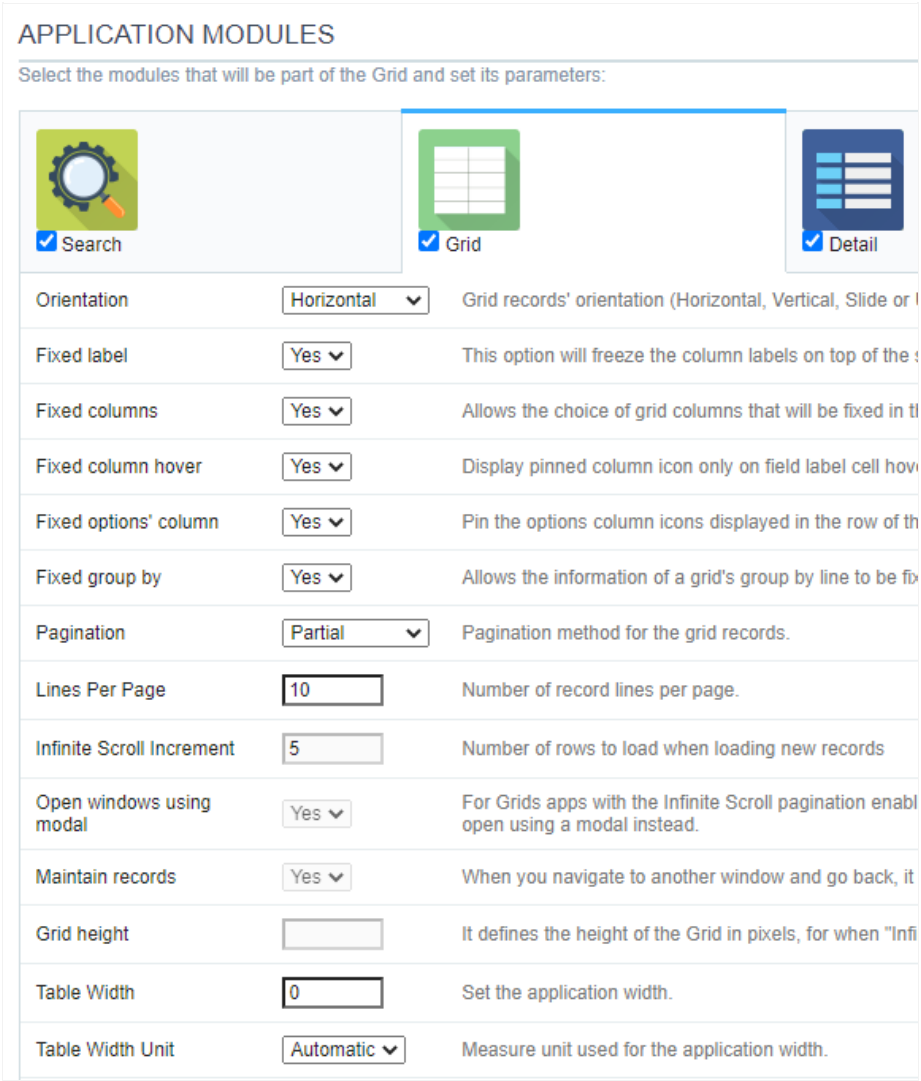
Grid

Search Settings

- **Filter Module** : This option configures which application will be called after the Search.
- **Use Iframe** : This option configures the Search to be displayed within an iframe (displaying the Filter itself and the search results on the same page). *This option is available only when the initial module is set to be the Search* .
- **Hide Grid Header** : Hide the Grid's header when using the Search in the iframe.
- **Show Results** : Display the search results on the same page when loading the application for the first time when the "Use Iframe" option is set as "Yes", otherwise, the results will be displayed only after the search.
- **Iframe Height** : Iframe height, in pixels, used to display the search results.
- **Search Modal** : This option configures the Filter Application to open in a modal window.
- **Search Modal Height** : Search Modal box height (in pixels). Option available only when the Search Modal box is enabled.
- **Search Modal Width** : Search Modal box Width (in pixels). Option available only when the Search Modal box is enabled
- **Table Width** : Filter application table width. This value can be in percent, pixel or automatic (set by Scriptcase). If you change the value you will also need to set the "Table Width Unit"
- **Table Width Unit** : This option configures the unit for the table width (percent, pixel or automatic). The value itself must to be set within the option "Table Width".

> You can customize more options from this module accessing the Advanced Search options. Learn More

Grid



Grid Settings

Orientation

This option configures the Grid records' orientation (Horizontal, Vertical, Slide or User Defined). When using the "User Defined" option you can design the HTML manually inside the option "Layout » HTML templates" and select within the Layout settings.

Fixed Label

This option will fix the columns' label at the top of the page (it is only available when the Grid records' orientation is set as Horizontal).

Fixed columns

Enables the option of fixing one or more columns of the query application during its execution. The resource will be represented by the icon positioned on the right side of the label.

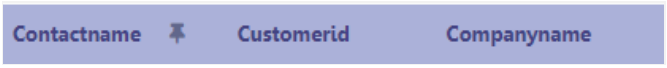
Icon color is defined by application theme.

When fixing a column, all others positioned on the left side will also be fixed.

For example, when fixing the column **Contact** the columns **Customer Code** and **Company** were also fixed.

In order for only the Contact field to be fixed, it is necessary to move it to the left corner using the **Columns** button.

See how configure the option of columns in the grid.



Fixed column hover

Defines if the fixed column button will always be visible in the application or only when the field hovers.

This hover option works only in applications running on the desktop, if the application is opened on a mobile device, the pin column icons will always be visible.

When checking yes the icon will be displayed when hovering the mouse over the column.

When checking no the icon to fix the column will always be visible.

Pin options column

Allows you to fix the options column displayed in rows of records.

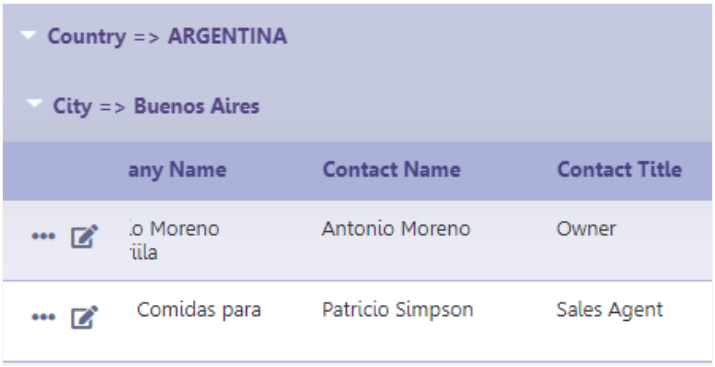
If the developer chooses to **mark yes** the column will remain fixed without the possibility for the application user to change its behavior, as in the image below.

When checking no the option to fix the column is at the discretion of the application user and the fix icon will be displayed, following the behavior defined in the configuration previous **Hover of fixed columns**.

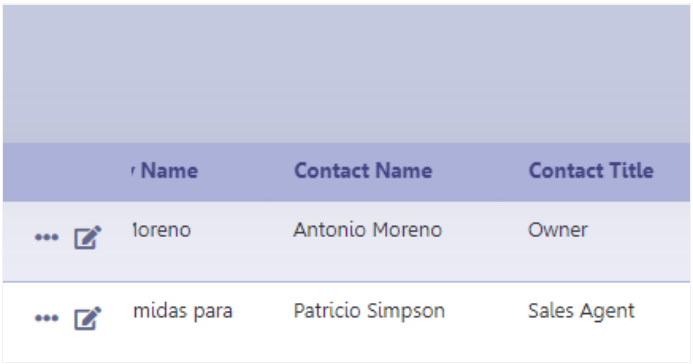
Fixed breaks

This option fixes the line with the break information when there is a horizontal scroll.

When checking yes if the application has a horizontal scroll, the break information will always be visible.



When checking no when scrolling horizontally, the break information is no longer displayed, as in the example below.



Pagination

This option sets the Grid paging type: Partial (pagination according to the amount of records set per page), Total (displays all records) or Infinite Scroll (automatic scrolling according to the increment)

Lines Per Page

This option sets the amount of records per page for the Grid

Infinite Scroll Increment

This option sets the number of rows displayed on each increment of new records. It is available only when paging is configured with Infinite Scroll.

Open windows using modal

For Grids application with the Infinite Scroll paging enabled, this option will set the links between applications and details of the Grid, that are configured to open in iframe, to open using a modal instead.

Maintain records

It configures if the amount of records displayed will be preserved when the application navigates to another window and go back.

Grid height

It sets the height of the query in pixels. If it is empty

Table Width

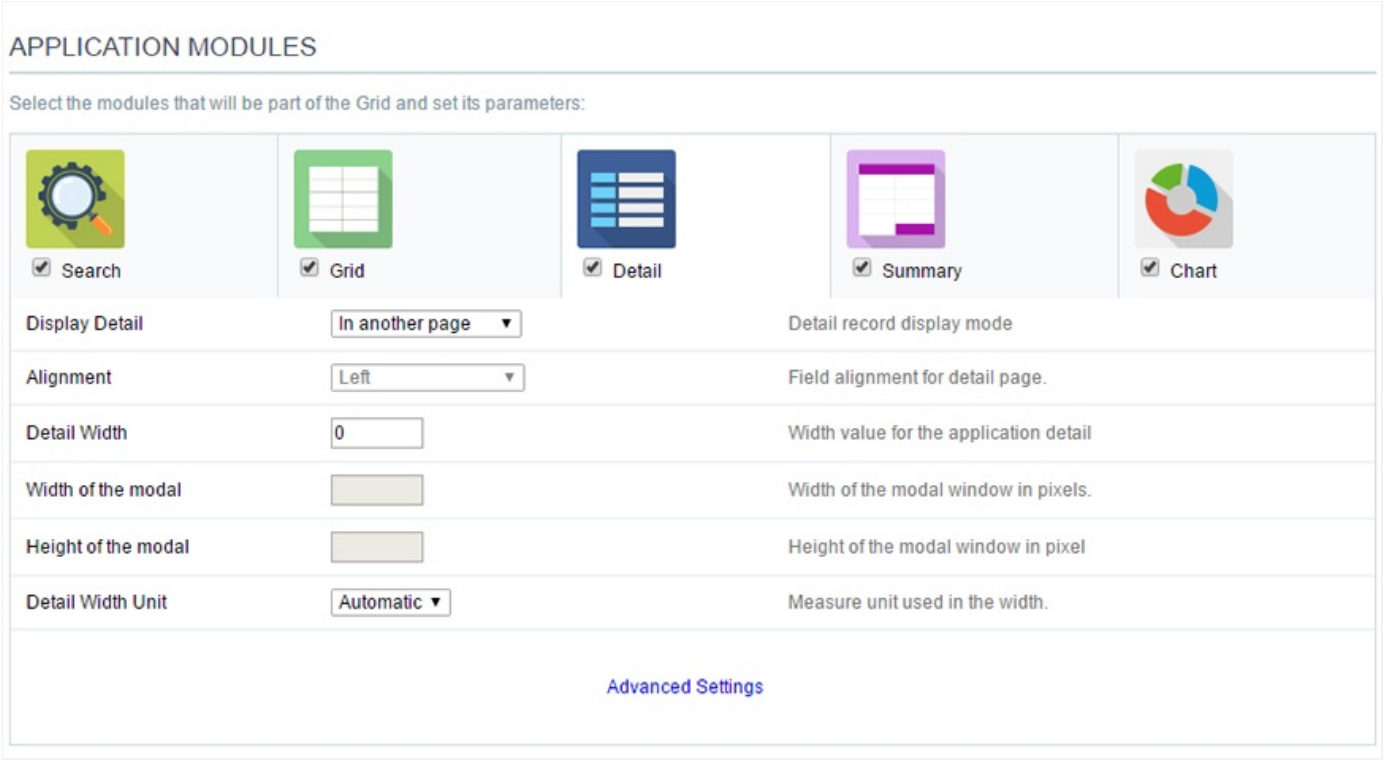
Grid application table width. This value can be in percent, pixel or automatic (set by Scriptcase). If you change the value you will

also need to set the “Table Width Unit”

Table Width Unit

This option configures the unit for the Grid table width (percent, pixel or automatic). The value itself must to be set within the option “Table Width”.

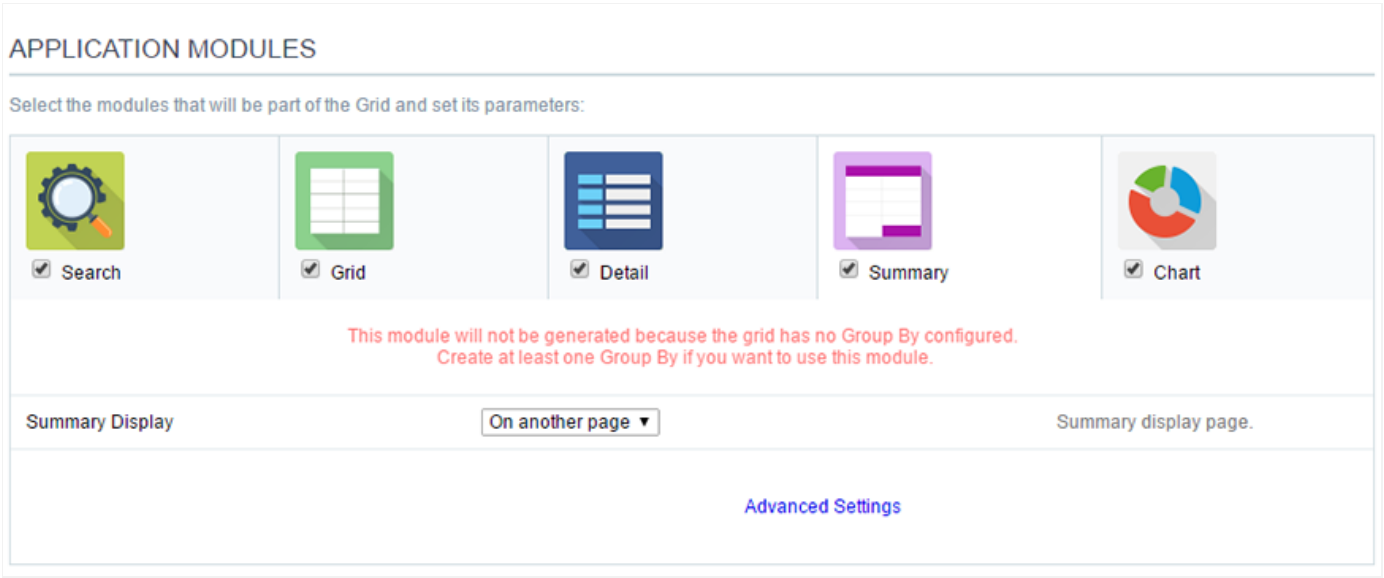
Detail



Grid Detail settings

- **Display Detail** : Allows you to set where the detail will open within the Grid Application * **Beside the Grid** : Displays the Grid Details to the right of the record, in the same window where Grid is being displayed. * **Below the Grid** : Displays the Grid Details below the Grid records, in the same window where Grid is being displayed. * **In another page** : Displays the Grid Details in another page, replacing the Grid view. * **In another window** : Displays the Grid Details in a separated browser window. * **Modal** : Opens a pop-up window to the display of the Grid Detail.
    - **Alignment** : Using this option you can set an alignment (center, right or left) for the Grid Detail when it’s using the “Display Detail” setting as “Beside the Grid” or “Below the Grid”.
    - **Detail Width** : This option sets the Grid Detail width when it’s using the “Display Detail” setting as “Beside the Grid”, “Below the Grid”, “In another page” or “In another window”.
    - **Width of the modal** : Allows you to customize the Grid Detail width when it’s using the “Display Detail” option as “Modal”.
    - **Height of the modal** : Allows you to customize the Grid Detail height when it’s using the “Display Detail” option as “Modal”.
    - **Detail Width Unit** : This option configures the unit for the “Detail Width” (percent, pixel or automatic).

Summary

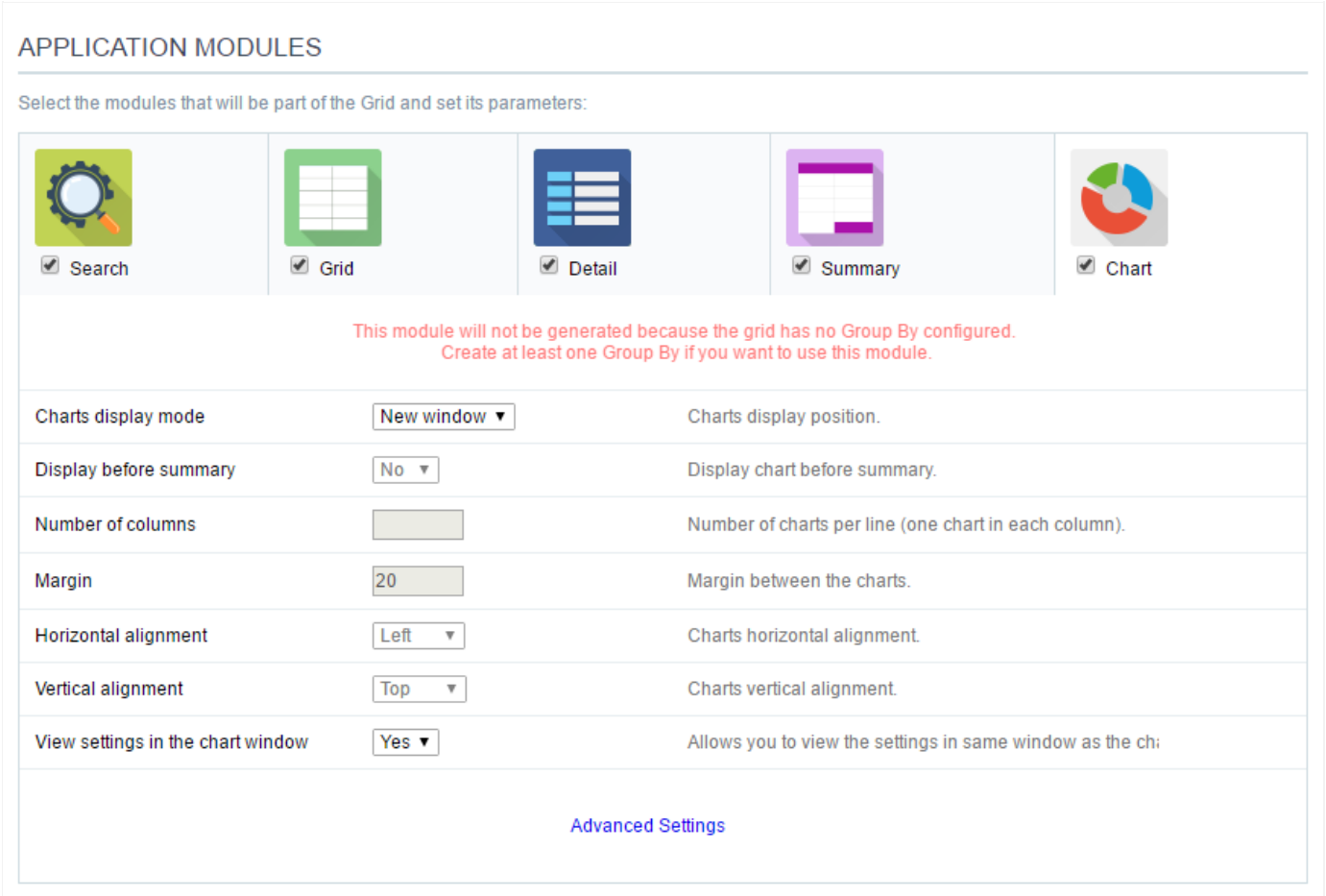


Grid

Summary Settings

- **Summary Display** : Sets the Grid Summary display option (On another page, On the last page, On every page). The Summary Application Module is only available when a group is created in the Group By settings.

Chart



Grid

Chart settings

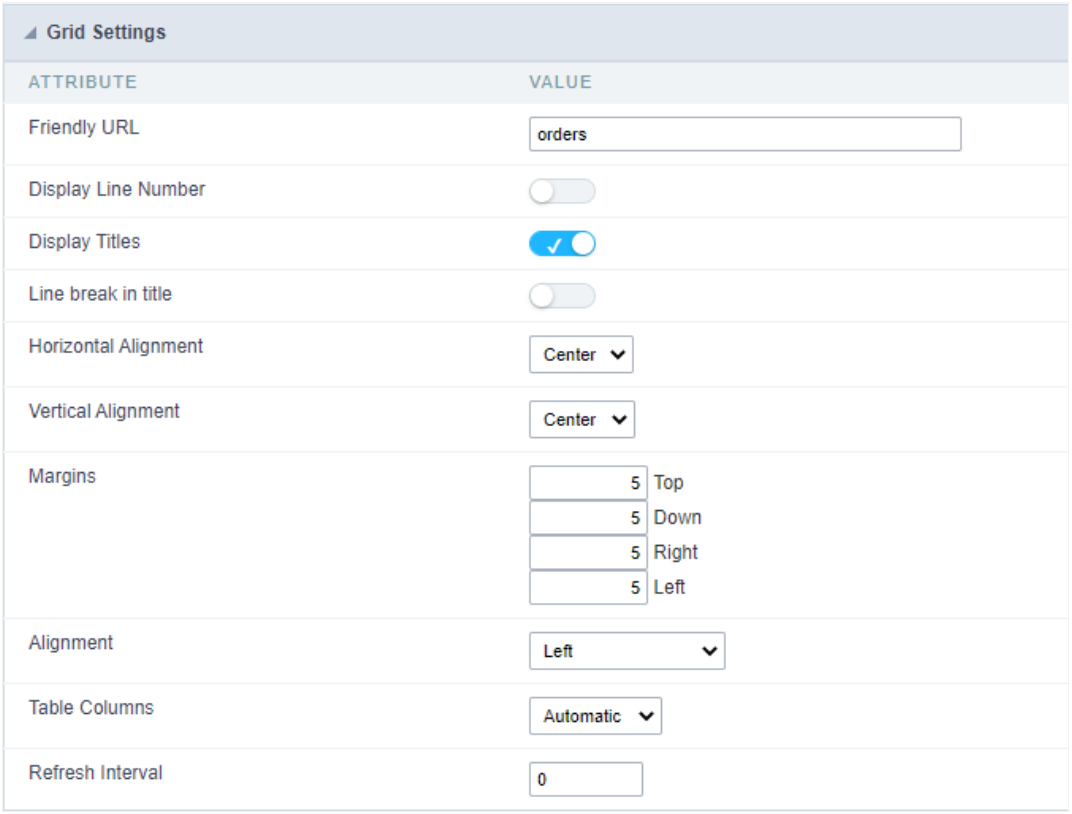
- **Charts display mode** : Sets the Chart Summary display option (New window, Same page, Other page).The Chart Application Module is only available when a group is created in the Group By settings.
- **Display before summary** : Sets whether the Chart is displayed above or below the Summary. Available when the Charts display mode is configured in the Same page.

- **Number of columns** : Sets the Chart number of columns per line (one in each column). Available when Charts display mode is configured in the Same page.
- **Margin** : Sets the Charts horizontal margin. Available when Charts display mode is configures in the Same page.
- **Horizontal alignment** : Sets the Charts horizontal position (Left, Right, Center). Available when Charts display mode is configures in the Same page.
- **Vertical alignment** : Sets the Charts vertical position (Top, Center, Bottom). Available when Charts display mode is configured in the Same page.
- **View settings in the chart window** : Sets the chart settings display on the same page. Available when the Char display mode is configured on Other page or new window.

Related Links 🔗

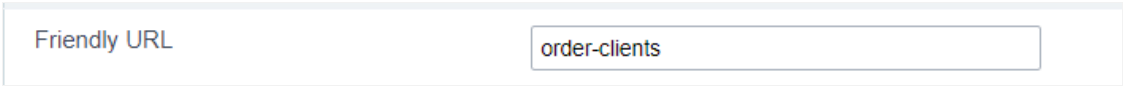
Related Video ▷

Grid Settings



Friendly URL

This attribute defines the application's friendly URL.



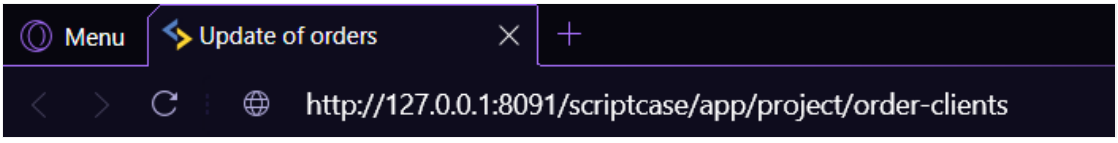
Alphanumeric characters and some special characters are allowed, such as: **hyphen** (-), **underscore** (_), **comma** (,), and **dot** (.).

The use of accents or spaces is not allowed.

Some recommendations for definition are:

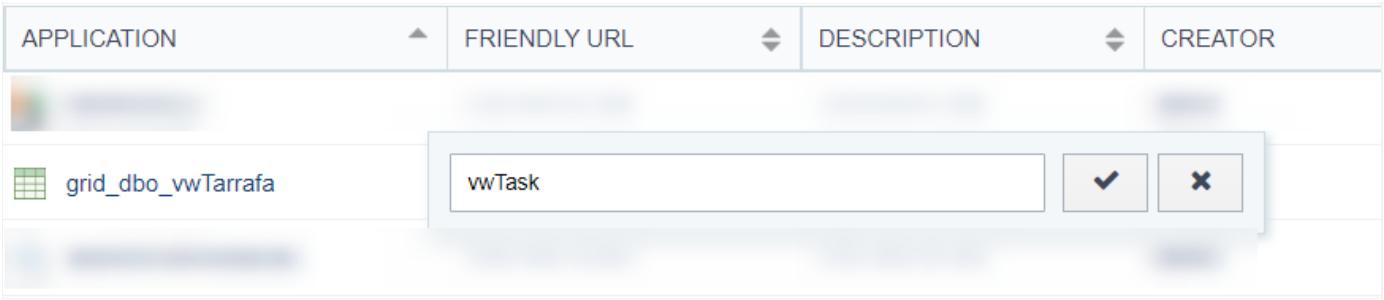
- Use keywords for identification.
- Use hyphens to separate words.
- Use only lowercase letters.
- Avoid using dates.

Example of an application using a Friendly URL



The friendly URL can also be defined in the application list on the project's initial screen.

See the example below

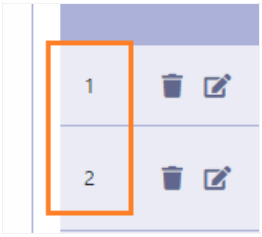


Display Line Number

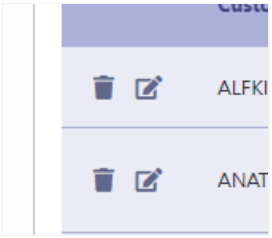
This attribute defines whether or not the line counter is displayed in the application. The row counter will be positioned next to the options column.

This feature is only available for forms configured in the formats: **Grid Editable** and **Grid Editable (View)**

Example with display line number enabled



Example with display line number disabled



- **Display Titles** : Allows you to set if the titles of fields will be displayed or not.

Line break in title

This attribute defines whether or not the line break in the field label will be allowed.

This attribute is only available in horizontal forms (Multiple Records, Editable Grid and Editable Grid (View)) configured to use **partial pagination**.

When enabled, does not allow line breaks in the field label. In this way, even if the field occupies a smaller space than the label, as in the Customer Code field in the image below, the line will not be broken.

Example of line break in title enabled

When disabled, allows a line break in the field label. When using this option, the space destined to the label will be defined by the size of the field, in this way, if the ladel is larger than the space destined to the field, there will be a break in the line.

Example of line break in title disabled

Customer identification	Employeeid
ALFKI	1
ALFKI	1

Horizontal Alignment

Defines the horizontal alignment of the application and can be defined as: **Centered**, **Left** or **Right**.

The attribute's default setting is **Centered**

Centered

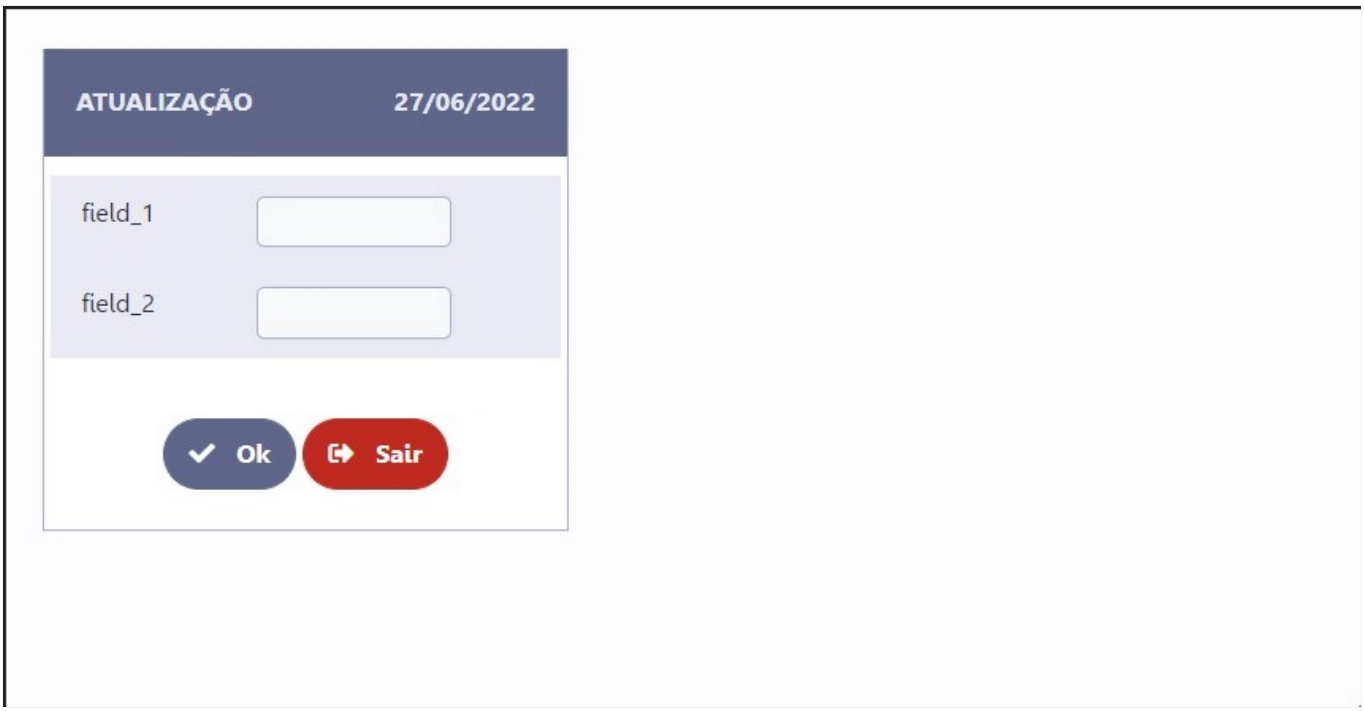
Positions the application in the center of the page, respecting the margin definitions defined in the theme or in the **margin** attribute.

Application example with centered alignment

Left

Positions the application in the left corner of the screen, respecting the definitions of the margins defined in the theme or in the **margin** attribute.

Left-aligned application example



Right

Positions the application in the right corner of the screen, respecting the definitions of the margins defined in the theme or in the **margin** attribute.

Right-aligned application example

Vertical Alignment

This attribute defines the vertical alignment of the application and can be configured with the following options: No Value, Top, Center ou Bottom.

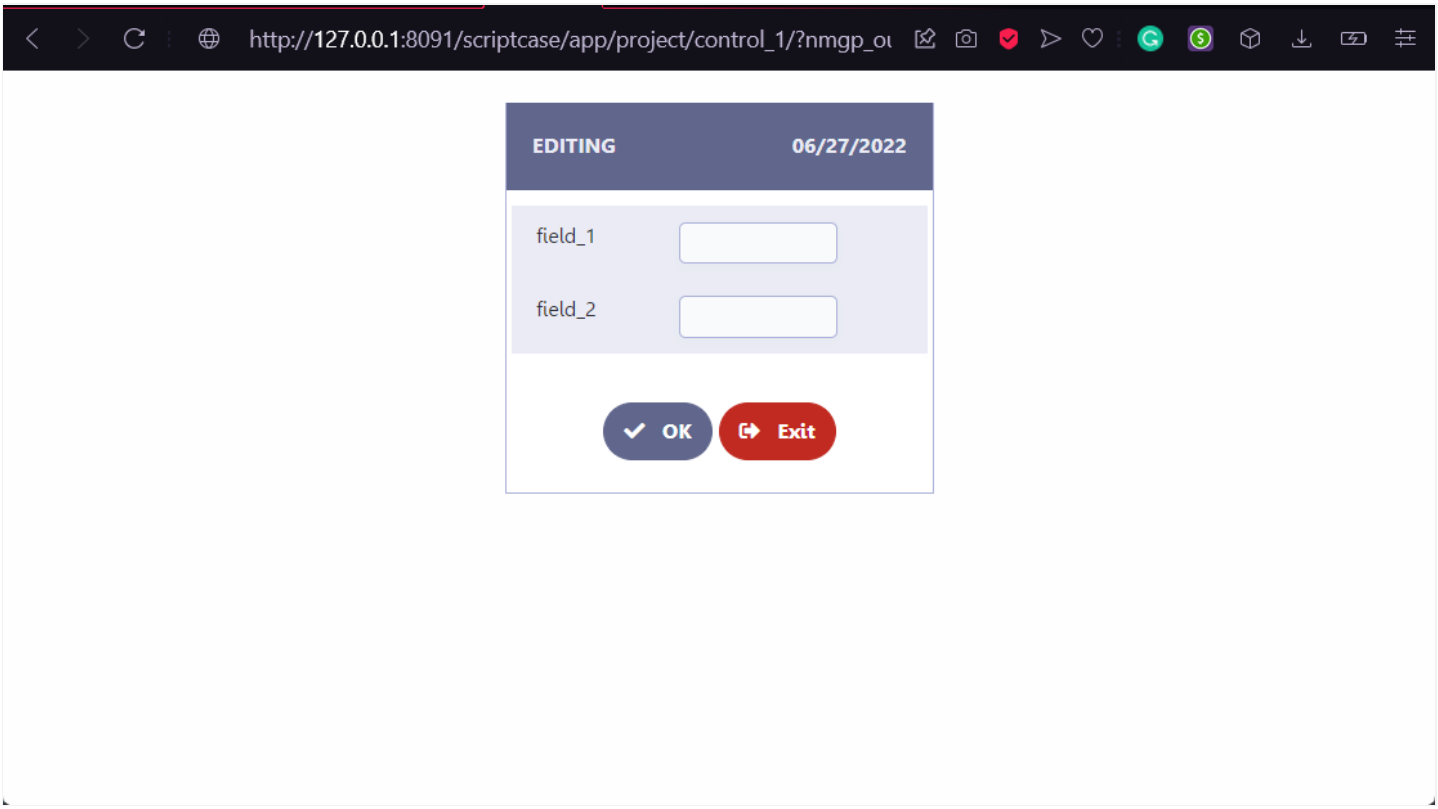
When using **Top**, **Center** or **Bottom** alignments, the margin attribute is disregarded.

See below for details on each of the alignment options.

No Value

Standard vertical alignment for new applications. In this configuration, the application is positioned at the top of the screen respecting the margins defined in the **margin** attribute or in the definition of the theme used in the application.

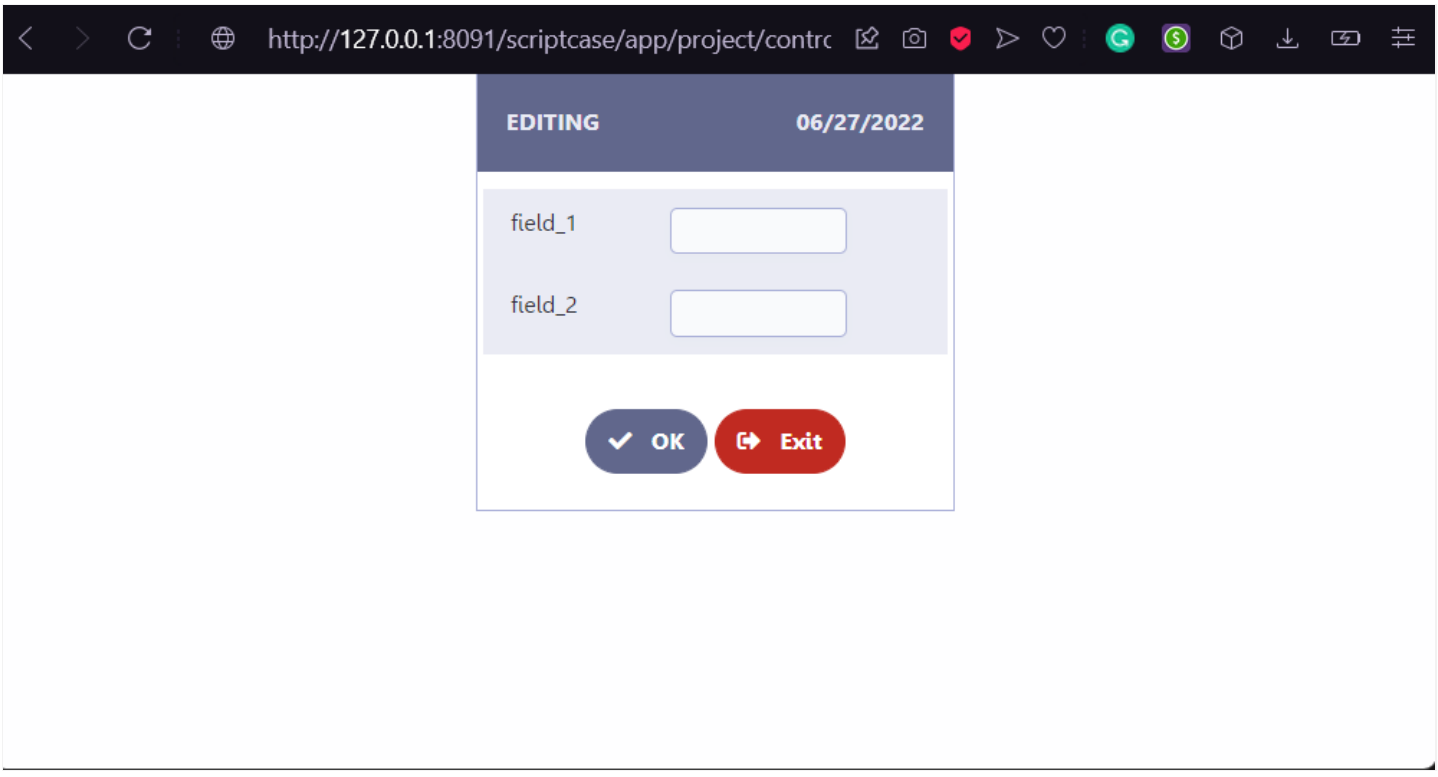
Application example with standard alignment



Top

Places the application at the top of the screen, disregarding the definitions of margins defined in the application theme or in the **margin** attribute.

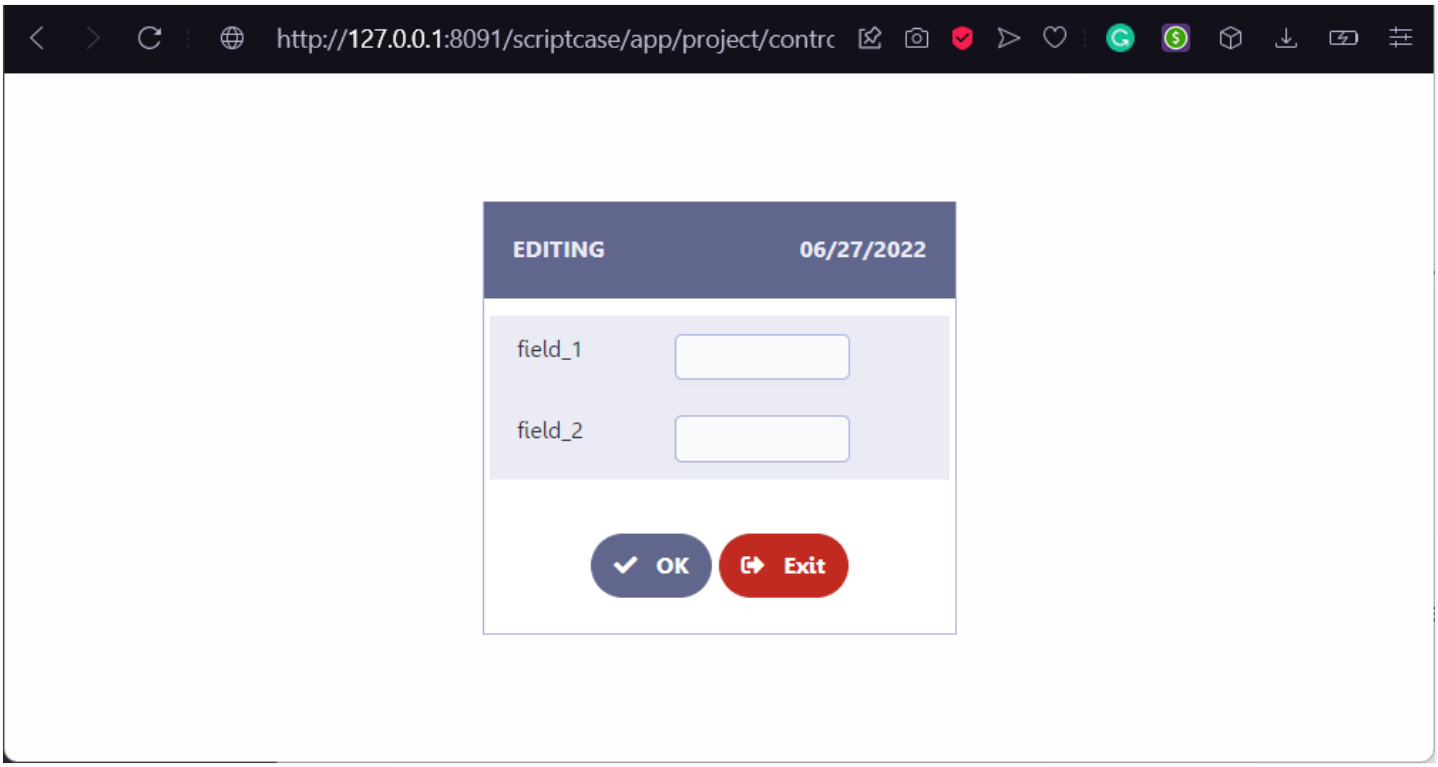
Application example with alignment on the top



Center

Positions the application in the center of the screen, disregarding the definitions of the margins defined in the application theme or in the **margin** attribute.

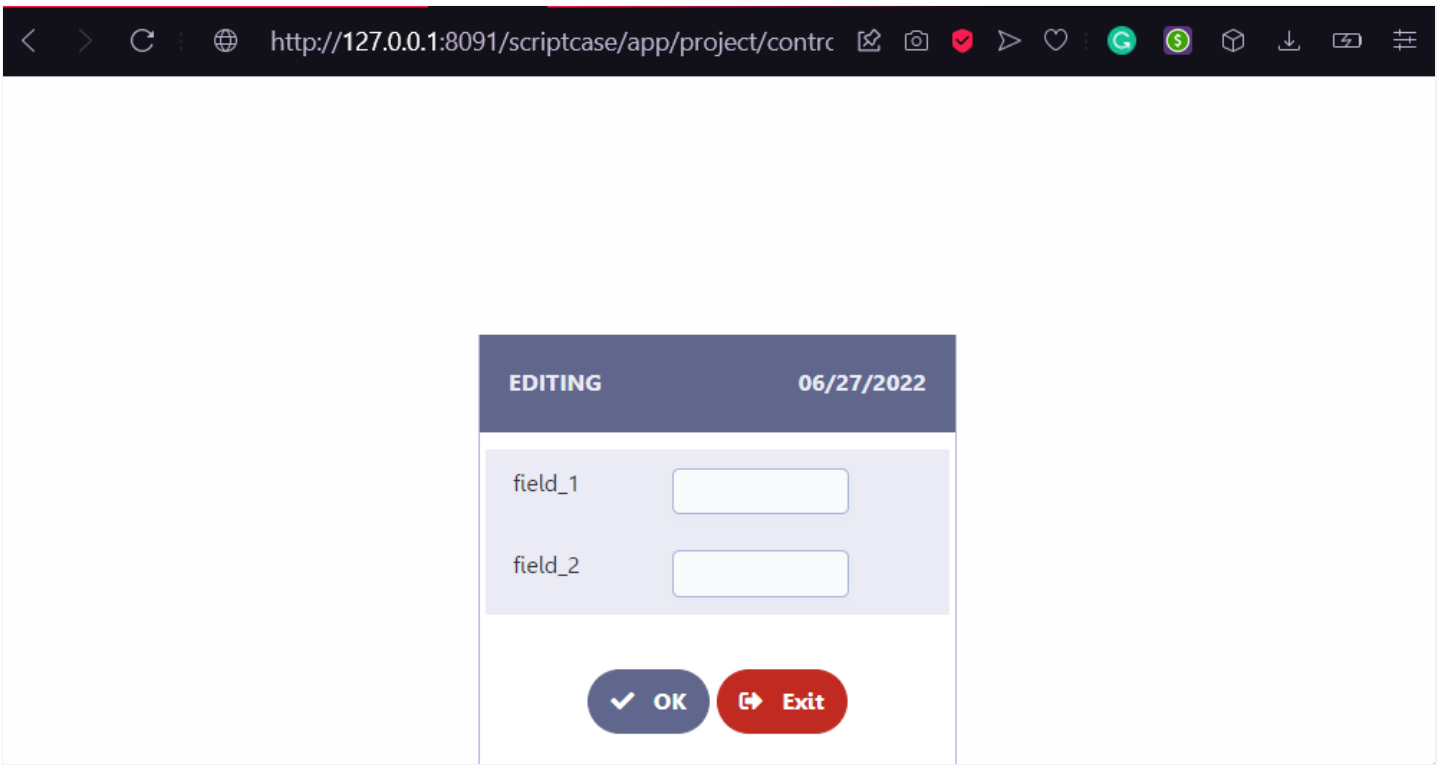
Application example with Center alignment



Bottom

Places the application at the bottom of the screen, disregarding the definitions of margins defined in the application theme or in the **margin** attribute.

Application example with alignment on the bottom

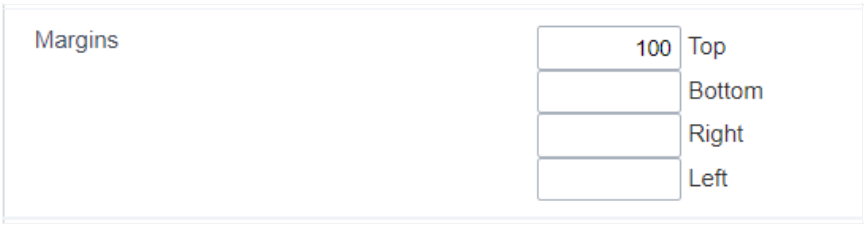


Margins

Allows the developer to set the application's margin values in pixels. If no value is specified, the application will use the default values from the configured theme.

Attention:
When configuring the **Vertical Alignment** attribute, the defined margin values will be ignored.

Example of how to use the attribute



Field alignment

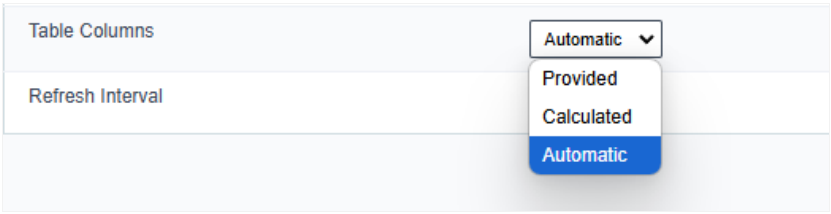
Defines the alignment of field values when the grid uses **Vertical** or **Slide** orientations.

This attribute is only considered in grids configured as Slide or Vertical

It allows defining the alignment of fields when the query orientation is configured as Vertical or Slide.

Table columns

This parameter defines the width of the application fields. The available options are: **Informed**, **Calculated** e **Automatic**.



Informed

This property defines that the width of the fields will be defined individually, at the field level, through the **Width** attribute.

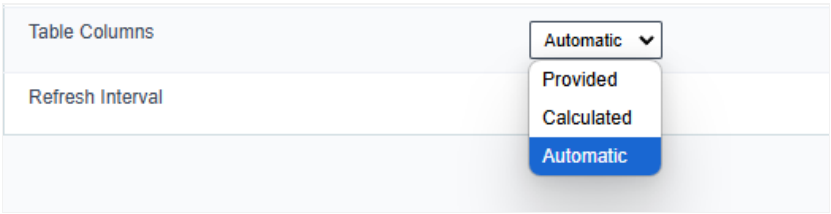
Using this property, it is mandatory that the width attribute in the field configuration has a value, otherwise, the field will assume the behavior of the **Automatic** property.

Calculated

The field width is calculated by our generator according to its field type and size.

Automatic

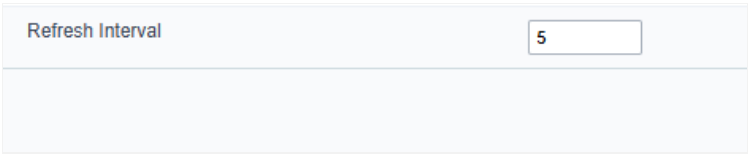
In this case the field width will be defined using browser criteria. No specific width for the fields will be informed.



Refresh interval

This attribute defines the page refresh time. This time is defined in seconds and only numbers must be informed.

By default the attribute is defined with **0**. With this value there will be no page reload.



Grid Edit Fields

This interface is useful for editing the field settings and their position to display.

Fields	Label	Datatype	Line Break	Line Break PDF	Title Horizontal Alignment	Text Alignment	Font Color
PAGE: PAG1							
BLOCK: GRID_ORDERS							
orderid	Orderid	Integer	☐	☐	Right	Right	
customerid	Customer Name	Text	☑	☐	Left	Left	
employeeid	Employeeid	Integer	☐	☐	Right	Right	
orderdate	Orderdate	Date	☐	☐	Center	Center	
requireddate	Requireddate	Date	☐	☐	Center	Center	
shippeddate	Shippeddate	Date	☐	☐	Center	Center	

Fields

Allows accessing the field settings (pencil icon on the left). You can change the field position by dragging them to the desired position. Drag a field to “fields not displayed” if you don’t want it in the app.

Label

Defines the title of a field in the app. For example: if the field name in the database is fld_txt_customer_name, you can display the label “Customer Name”.

Data Type

It informs the data type of the field.

Line Break

Allows the Line Break in the records when the field text is greater than the column width.

Line Break PDF

Allows the Line Break when the user exports the app as a PDF.

Title Horizontal Alignment

Defines the horizontal alignment of the field label.

Text Alignment

Defines the horizontal alignment of the field text.

Font Color

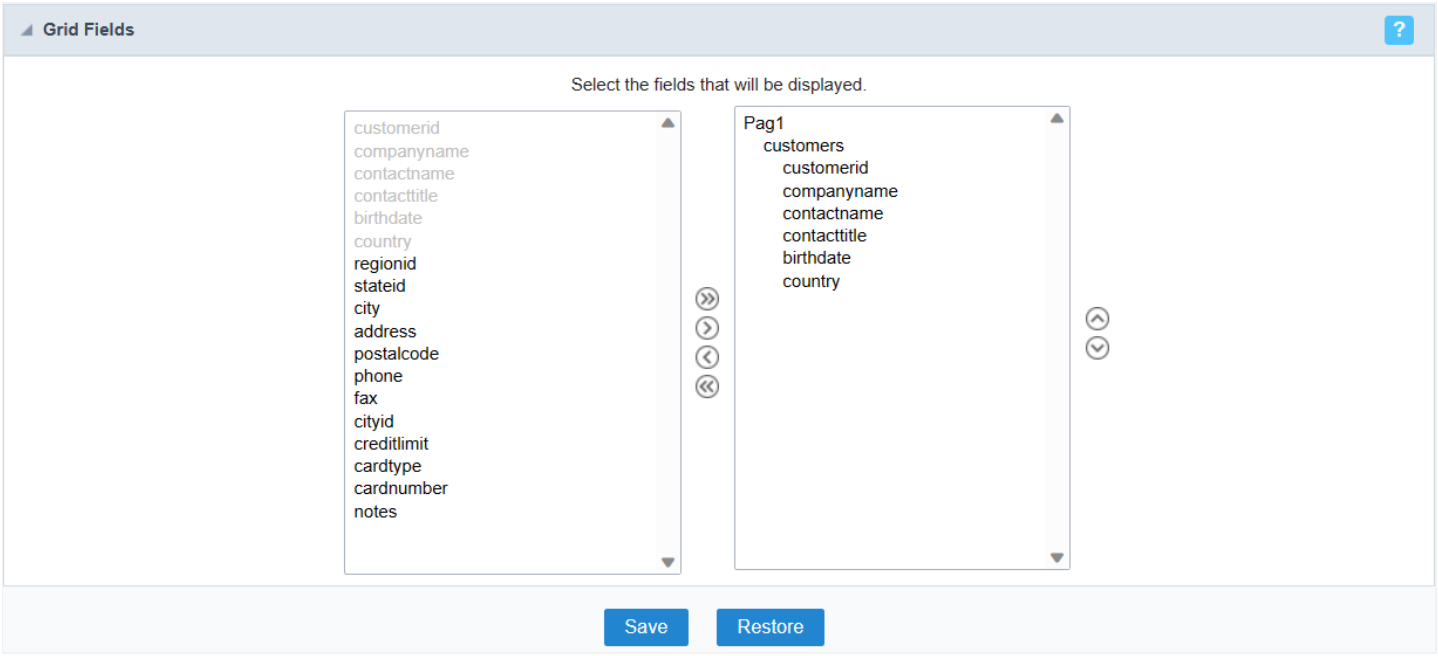
Configure the text color for the field.

Grid Fields Positioning

In this configuration screen we have a list of all the fields available in the application, whether they are fields mapped from the table or virtual fields (Created only in the Scriptcase interface).

It is also possible to allow the end user to manipulate the application's fields in the way they prefer, for that we must add the **columns button** in the application's toolbar.

In the **left column**, we have a list with all the fields available for use and in the **right column** we have a list of the fields that were selected to be displayed in the running application.



Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

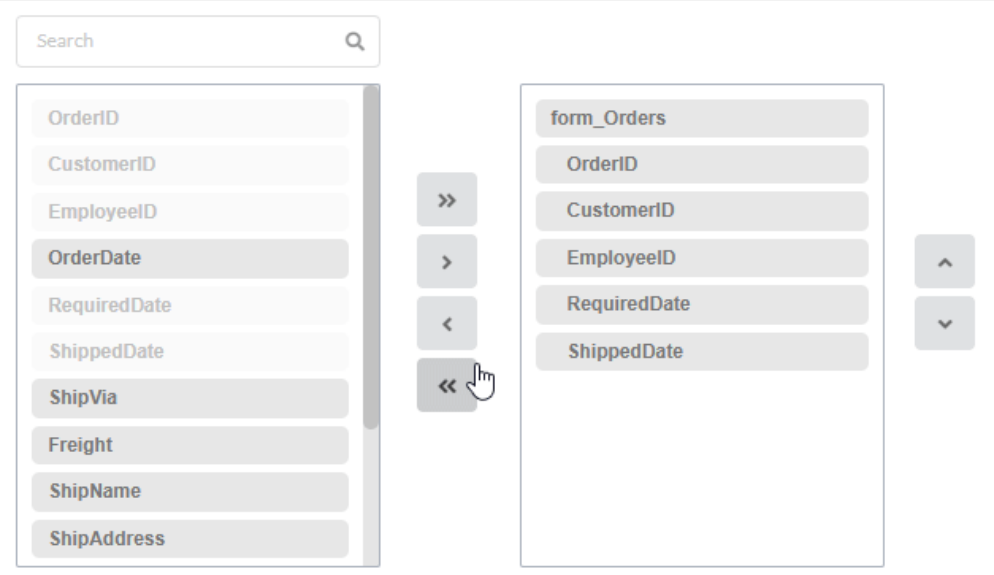
- » - Move all fields to the right.
- › - Moves only selected fields to the right
- ‹ - Moves only the selected fields to the left.
- « - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the › button.

To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning



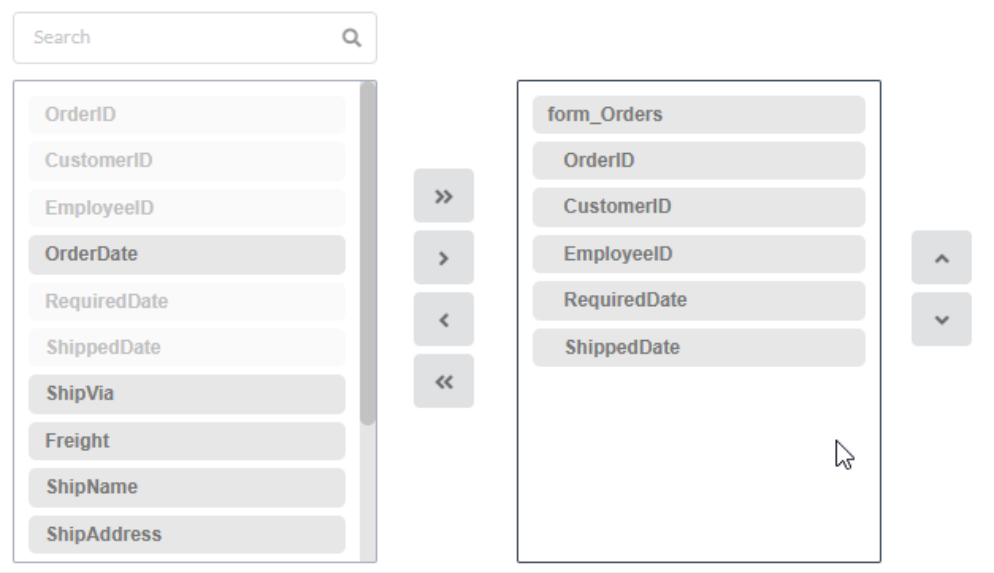
Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

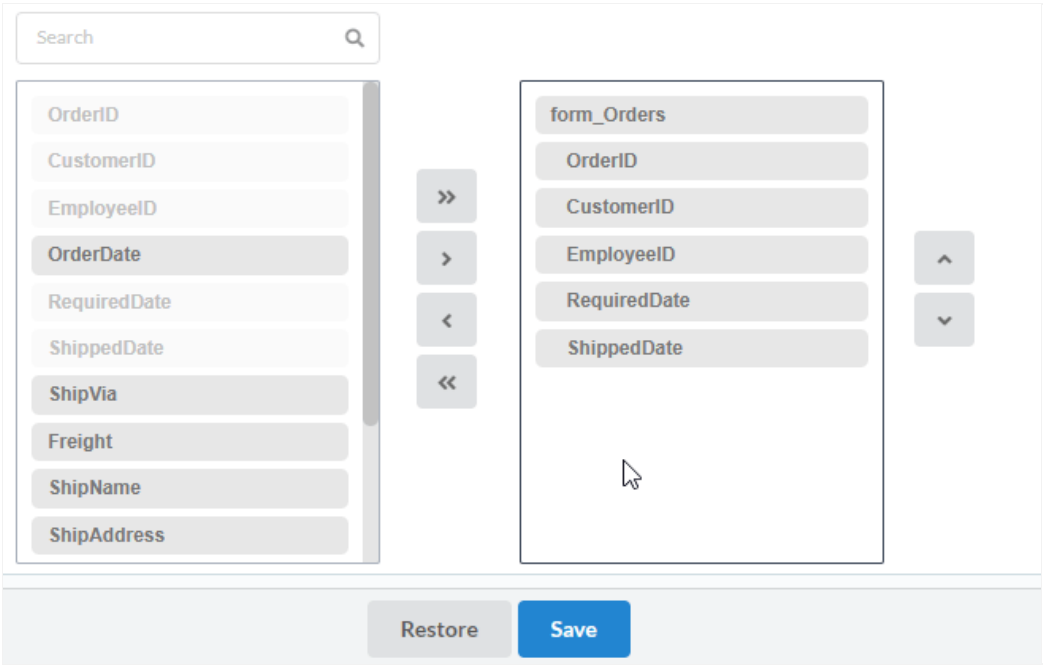


Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.


Search
OrderID
CustomerID
EmployeeID
OrderDate
RequiredDate
ShippedDate
ShipVia
Freight
ShipName
ShipAddress
form_Orders
OrderID
CustomerID
EmployeeID
RequiredDate
ShippedDate
Restore
Save

Grid Toolbar

The application toolbar has two segments: Top and Bottom, in a way that is possible to define to display buttons into both areas. Those areas work independently, allowing them to display the same button, for example.

It's also possible to select the buttons and their position if the application is running on a mobile device.

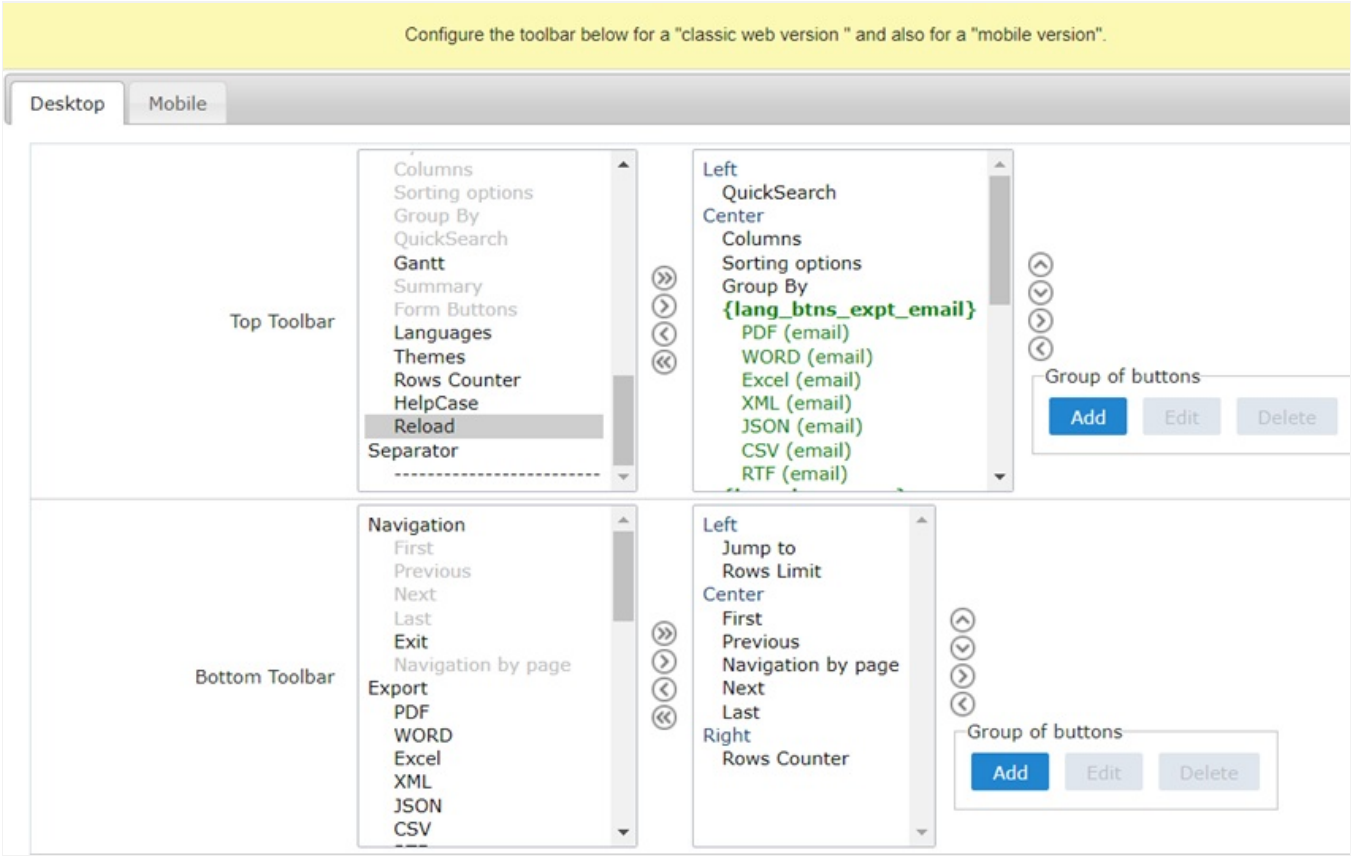
Toolbar

Desktop

Here we must inform the toolbar settings for the "Classic Web Version" mode and which buttons are available in the application when accessed from a **Desktop** environment.

Mobile

Here we must inform the toolbar settings for the "Mobile Version" mode. That is which buttons are available in the application when accessed from a **Mobile** dispositive.



Navigation:

Buttons relative to the navigation of the application.

Next	Move to the next page that can be a single record or a list of records.
Previous	Returns to displays the previous page records or a single record.
First	Move to the First page or record
Last	Move to the Last page or record
Exit	Close the application
Navigation by page	Displays a “page-number” navigation bar. **Example:** 1 2 3 4 5
Reload	Displays a button to reload the query data

Export:

Groups the options relative to the generated exports. Scriptcase generates the following export formats:

PDF	Generates a complete Report with all the data of the application in a PDF format.
WORD	Generates a complete Report with all the data of the application in a WORD format.
XLS	Generates a complete Report with all the data of the application in an EXCEL format.
XML	Generates a complete Report with all the data of the application in an XML format.
CSV	Generates a complete Report with all the data of the application in a CSV format.
RTF	Generates a complete Report with all the data of the application in an RTF format.
Print	Creates an HTML with the records ready for printing.

Export by Email:

Groups the options relative to the emails exports. Scriptcase generates the following export formats:

PDF (Email)	Sends by Email a complete report with all the data of the application in a PDF format.
WORD (Email)	Sends by Email a complete report with all the data of the application in a WORD format.
XLS (Email)	Sends by Email a complete report with all the data of the application in an EXCEL format.
XML (Email)	Sends by Email a complete report with all the data of the application in an XML format.
CSV (Email)	Sends by Email a complete report with all the data of the application in a CSV format.
RTF (Email)	Sends by Email a complete report with all the data of the application in an RTF format.

Others:

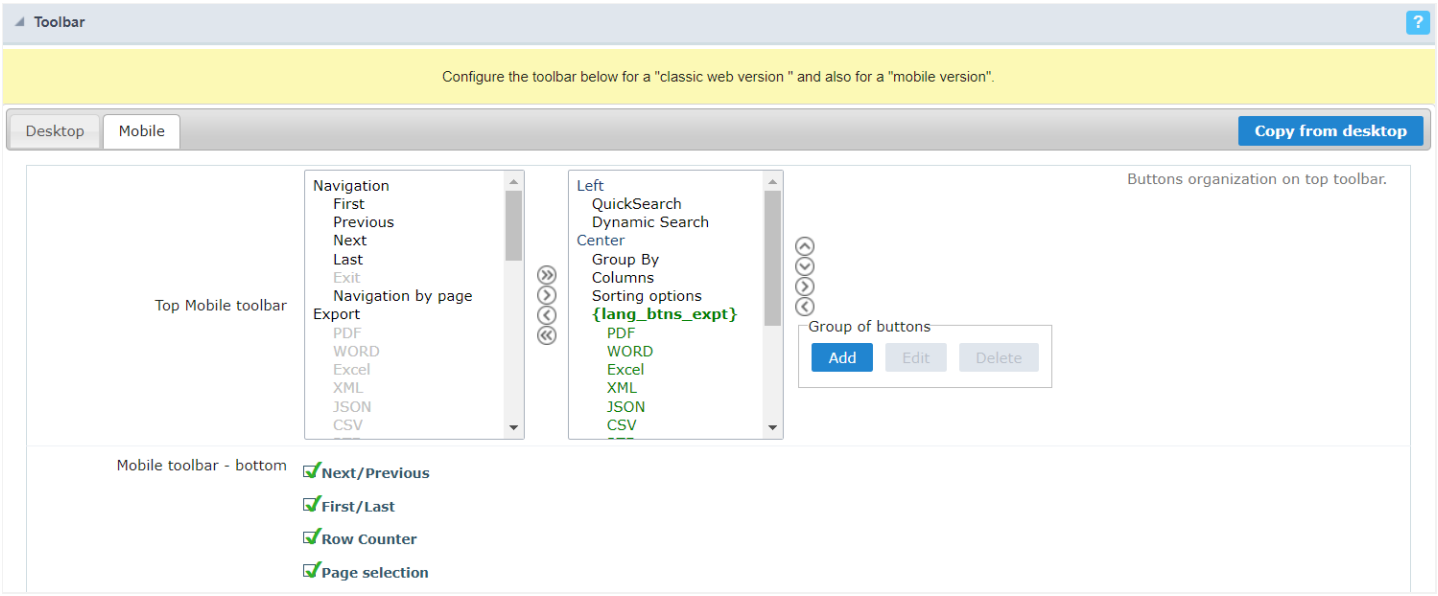
Other options available in the Grid application.

Jump to	Move to the informed page.
Rows Limit	It is a Combobox that defines the number of rows per page.
Search	Goes to the Search Form to filter the records.
Dynamic Search	It displays the fields of the search to filter the records.
Columns	Allows to include or remove columns of the Grid on the fly.
Sorting Options	Allows selecting the order of the records based on the field.
Group By	Allows to select or change a Group By rule on the Grid.
Save Grid	Allows saving the current state of the application. For example, in the advanced search, you can save the search data for further use.
Quick Search	Allows to perform a quick search in the records of the application.
Gantt	It displays a Gantt chart, if it was previously set.
Summary	It displays a summary with the synthetical data of the records.
Form Buttons	It displays the Form Buttons when there's an Application Link from the Grid to a Form.
Languages	It displays a Combobox with the names available, defined in the project properties.
Themes	It displays a Combobox with the themes available, defined in the project properties.
Rows Counter	It displays the number of records retrieved by the application.
HelpCase	Displays a button to open the help page.

Separator:

————————	Displays a line separating the buttons, when used the Group Buttons.

Toolbar Mobile



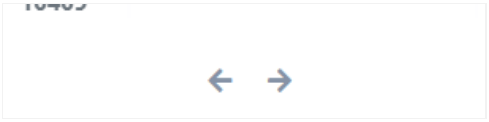
Top Mobile toolbar

It has the same options as the **Desktop** version, adding only the item “Copy from desktop”, which, when clicked, makes a copy of the items from the upper toolbar of **Desktop** to **Mobile**.

Mobile toolbar - bottom

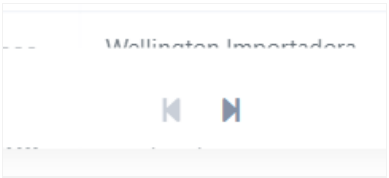
Next/Previous

Enables navigation to the next and previous page on mobile devices.



First/Last

Enables first and last page navigation on mobile devices.

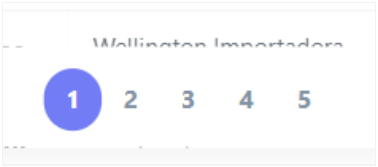


Row Counter

Enables the record counter showing the application’s total records

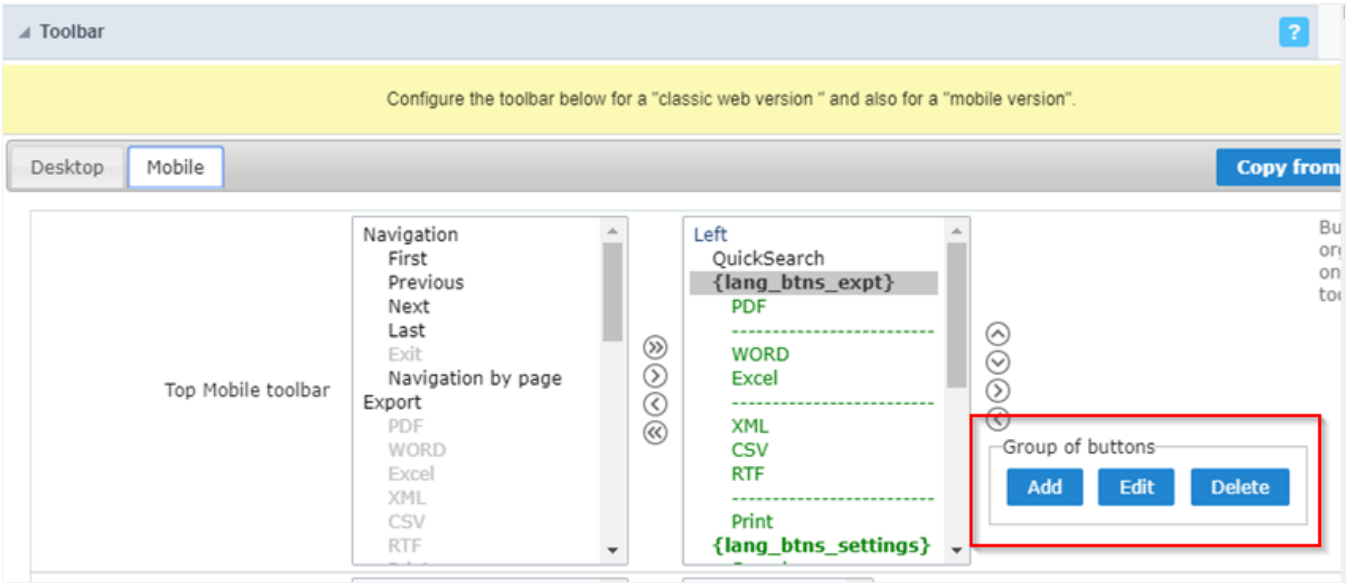
Page selection

Enables page navigation on mobile.



Buttons Group

The **Group** option allows you to group a set of buttons of the application toolbar to display them as a dropdown, for example.



Add

Add a new group of buttons.

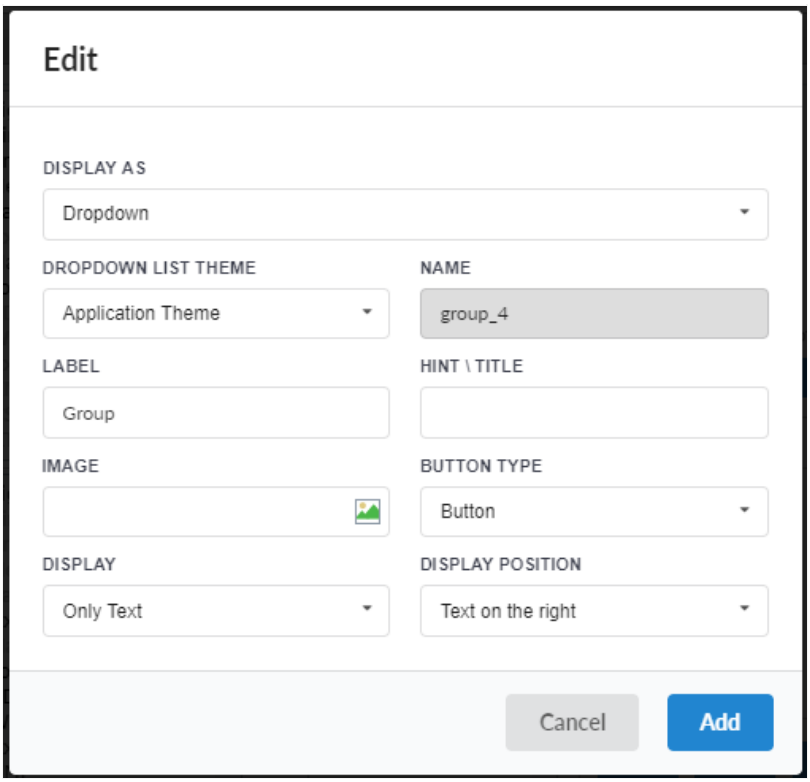
Edit

Edit an existent group of buttons.

Delete

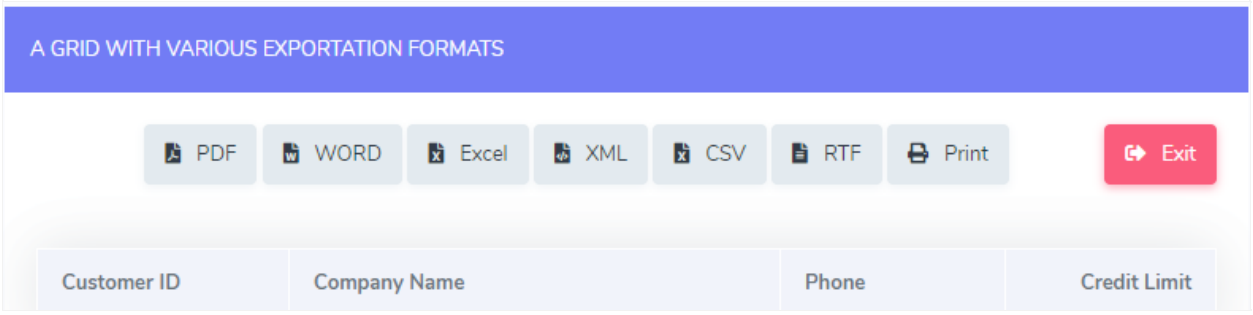
Delete the selected group of buttons.

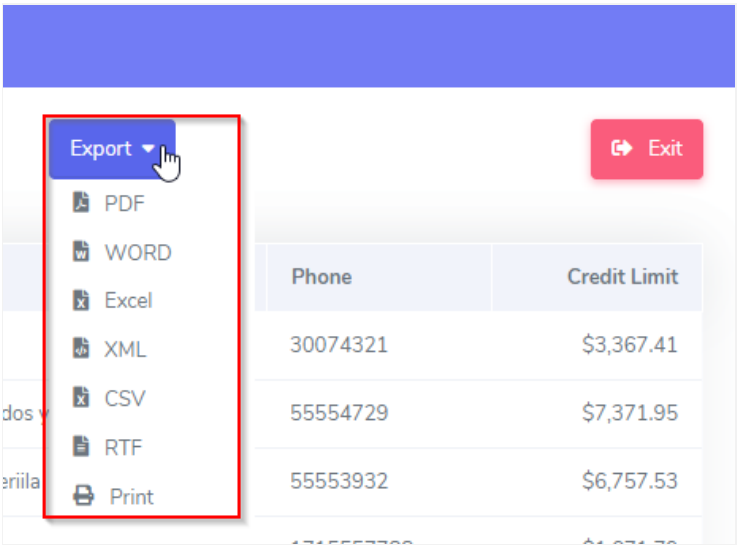
When you press the **Add** or **Edit** option, you can see the settings to configure the grouper:

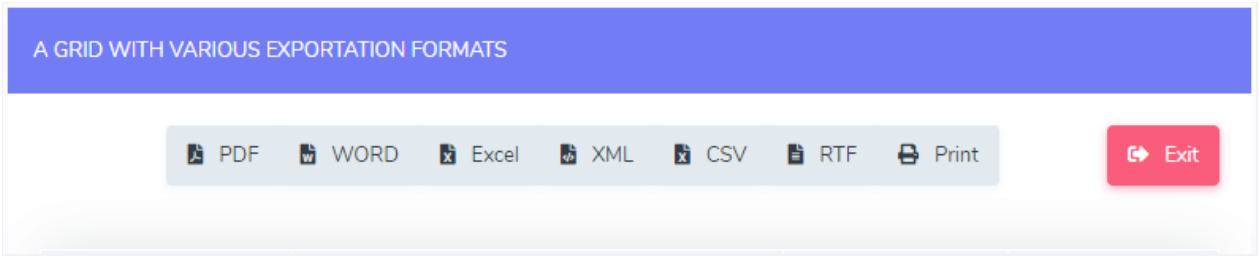


Display As

Allows displaying the group button as **Dropdown** or **Side by Side**.







Dropdown List Theme

Allows defining the Dropdown theme selecting between **Application theme** and **Button theme**.

Name

Allows defining a name for the button group.

Label

It is the displayed name for the button group in the application.

Hint\Title

Displays a hint to the end-user when the mouse is on the group of buttons.

Button Type

Allows displaying the button group as a Button, Image, or Link.

Image

Allows selecting an image for the button.

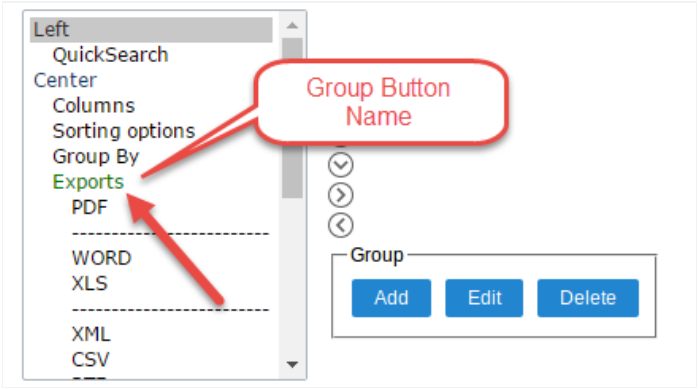
Display

Defines if the button displays only Text, only image, or both.

Display Position

Defines the position of the Text or Image (Text to the right, Image to the right).

After creating a button group, you need to move the grouped buttons below of the Button Group and then move them to the right. Like the image below:



Button Settings

Button	Label	Hint
QuickSearch		
Dynamic Search		
Insert		
Cancel		
Update		
Delete		
Exit		
Jump to		
First		
Previous		
Navigation by page		
Next		
Last		
Rows Counter		
New		

Button:

It displays the buttons available in the application.

Label:

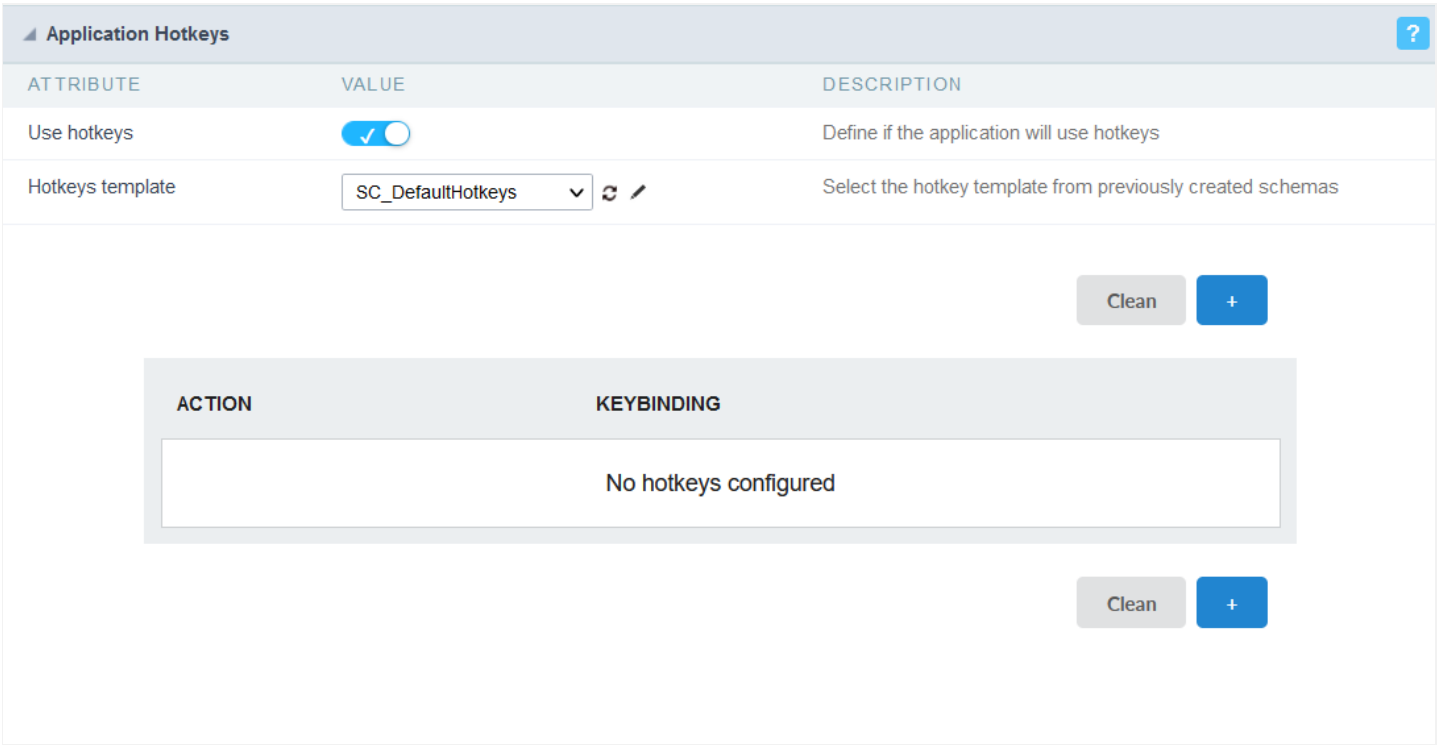
Allows defining the label of the buttons to display for the users.

Hint:

Allows defining the buttons hint to display for the users.

Application Hotkeys

Scriptcase allows creating shortcut keys to your applications. You can select a predefined template or create specific actions for an application.



Use hotkeys

Defines if the application uses hotkeys. When you enable this option, the default shortcut keys settings are disabled.

Hotkeys templat

Select the hotkey template previously created.

Action

Selects the triggered action when pressing the selected key.

Keybinding

Selects the keys responsible for executing the chosen action.

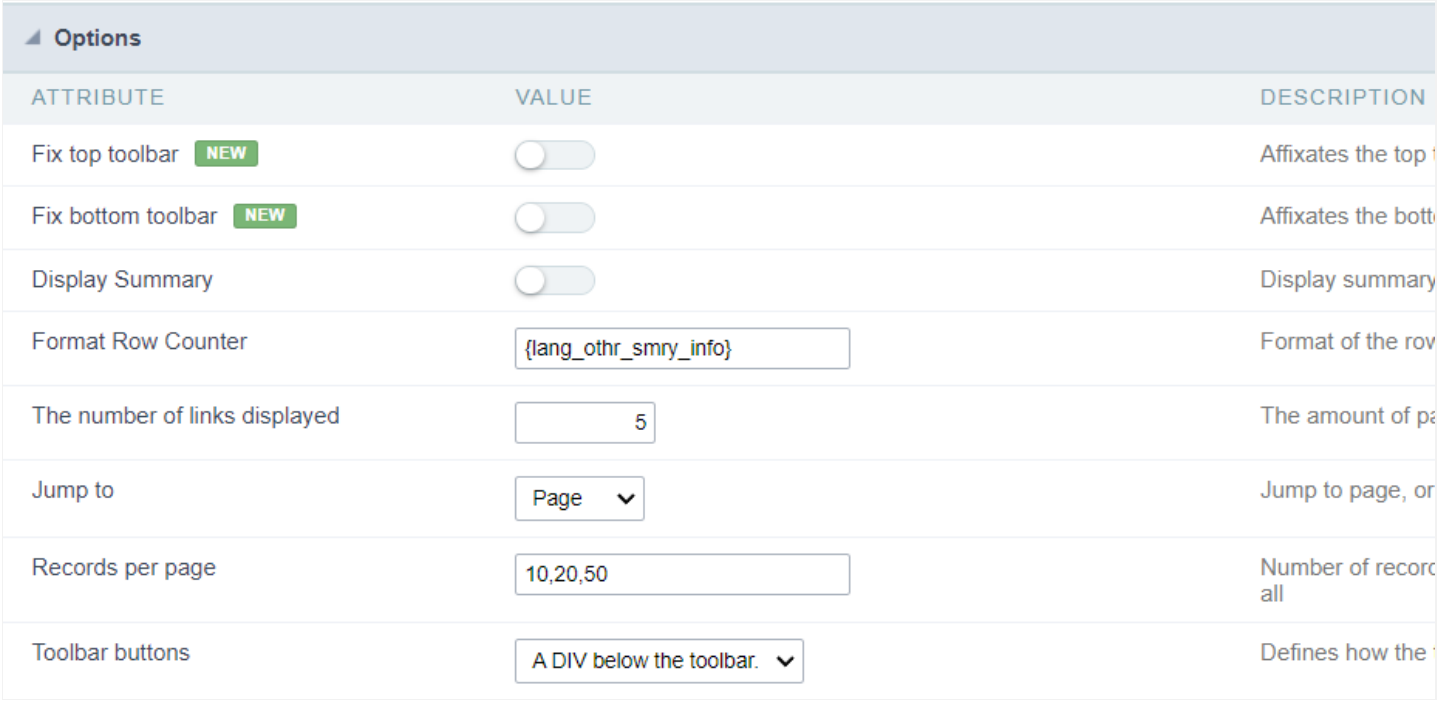
Add “+”

Adds a new action on the keys list.

Clear

It clears the selected hotkeys preference.

Options



Fix top toolbar:

Affixates the top toolbar to the top of the window

Fix bottom toolbar:

Affixates the bottom toolbar to the bottom of the window

Display Summary:

Allows to display the summary button or not.

Row Counter:

Allows to display the Row Counter or not.

Example: (1 to 10 of 200)

The Number of Links Displayed:

Defines the number of links per page, when the navigation option is disabled.

Jump To:

Move to the informed record page.

Records by page:

Allows defining the number of records to be exhibited for each page. To show all records use the option “all”.

Example: 10,20,30, all

Toolbar Buttons:

The type of view for the Toolbar buttons (A DIV below the toolbar, Modal).

Related Links 🔗

Related Videos ▷

Grid Export Overview

The Export menu allows the configuration of the various types of export present in the grid application.

The files generated in the export can only be viewed in the browser, downloaded to the client's machine or sent by email using the sending APIs (SMTP, Chuck or Amazon SES) built into the tool.

Each type of export has its own configuration that can be previously defined by the developer or passed on to the end user through the configurable export option, thus making the report generation more dynamic.

Grid Export Type List

PDF	XML
WORD	JSON
CSV	Impressão HTML
Excel	RTF

Related Links

- Export to PDF
- Export to Excel
- Export to JSON
- Export to Word, XML, CSV, Print and others
- Export Setting
- Export Summary to Email
- Export Applications
- Export Nested Grid to XML
- How to generate PDF reports with the export option
- Macro sc_set_pdf_name
- JSON exportation for grids and charts

Related Videos

Grid Export Email

In this menu, the service used to send emails will be defined in the **Sending Settings** block, as well as the body of the emails that will be sent in the **Export Settings** block.

> To use this resource, it is necessary that the fields for exporting by e-mail are enabled in the application's toolbar.

Sending settings

In this block we must define the sending service that will be used in the application. This service must be defined in the **API** attribute by selecting a Pre-configured API or selecting the **Custom** option that must be configured in the application.

> This option allows you to select only one delivery service for the application. For the dynamic definition of the service it is necessary to use the macro sc_send_mail_api.

API

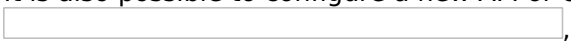
This attribute defines the service that will be used by the application to send emails.

The developer chooses between selecting one of the listed Email sending APIs or the **Custom** option that allows configuring a sending service in the application itself.

Pre-configured APIs

If there are one or more email APIs configured and active, they will be listed and categorized according to the **access level** at which they were created.

> **Access Levels**
>
> - **Public**: Available to all users on all projects.
> - **Project**: Only available in the project in which it was created.
> - **User**: Only available to the user who performed the API configuration.

It is also possible to configure a new API or edit an existing one by clicking on the icon  , that opens API configuration screen.

The icon is responsible for updating the list of APIs in the attribute by refreshing the field, allowing the new configured APIs to be displayed without the need to reload the page.

Custom

When selecting this option, the configuration fields will be displayed below the API Attribute, in this case, it is not necessary that there is an API configured in the **Tools > API** menu.

The developer must configure the sending service using the displayed fields according to the selected **Gateway**.

Scriptcase connects to three different APIs for sending emails: SMTP, Mandrill, and Amazon SES.

SMTP Configuration

SMTP is the standard protocol for sending e-mails over the Internet, and each provider has its SMTP.

> For more information about SMTP settings click here

API

It allows you to select an API already configured in **Tools > API** or set a new one here by selecting **- custom -**. In this case, you see some options according to the selected Gateway.

> To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

Gateway

Select the way to send the email. The SMTP is the default protocol for sending e-mails over the Internet, and each provider has its SMTP.

SMTP Server

SMTP server address for the provider.

Port SMTP

Port used by the mail server. Use port 465 for security with SSL, port 587 for security with TLS, or port 25 as port without security. By omitting the value, Scriptcase defaults to 25.

User SMTP

SMTP server user.

Enter SMTP

SMTP server user password.

Protocol SMTP

Defines the security protocol. By omitting the value, Scriptcase uses the default value.

E-mail

Enter the origin email, that is the email sender.

Name

The sender’s name displayed in the email.

Mandrill Configurations

Mandrill is a transactional email API for MailChimp users, ideal for sending data-driven emails.

API

It allows you to select an API already configured in **Tools > API** or set a new one here by selecting **- custom -**. In this case, you see some options according to the selected Gateway.

> To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

Gateway

Select the way to send the email. **Mandrill** is a transactional email API for MailChimp users, ideal for sending data-driven emails.

API KEY

Enter the key obtained from the origin site of your API.

E-mail

Enter the origin email, that is the email sender.

Name

The sender’s name displayed in the email.

> For more information about mandrill

Amazon SES Configurations

Amazon Simple Email Service Amazon SES is an email sending service designed to assist in sending marketing emails, notifications, and transactional messages.

> To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

API KEY

Enter the key obtained from the origin site of your API.

API Secret

Enter the secret access key of your account.

Region

Amazon SES has endpoints in many countries, to reduce network latency, inform the region of the endpoint closest to your application. See the regions.

E-mail

Enter the origin email, that is the email sender.

Name

The sender’s name displayed in the email.

> For more information about Amazon SES

Export Settings

It allows the definition of the default values of the interface for sending e-mails used by the end user of the application.

To

Set the default value for the field in the application’s end-user email sending interface.

Whether or not to inform a default value for the field is the choice of the developer, who can choose to activate the **empty attribute**, use a **global variable** or inform a **fixed text**.

> Whether using the variable or fixed text in the field, if you choose to inform more than two e-mails, they must be separated by **,** or **;**.

Empty attribute

When choosing not to inform a value, the developer forces the user to inform one or more destination e-mails.

Global variable

In this case, just add the variable to the value of the attribute containing the field’s default email list.

fixed text

The developer can define one or more default emails directly in the attribute without using a variable.

Example of displaying the (To) field in the email sending user interface

Below is an example of how the emails defined by the developer will be displayed on the sending screen to the application user. In this screen the user can remove the default values previously defined by the developer by clicking on **X** or can add new emails to the field.

It has copy

Defines the behavior of the **Copy (cc)** attribute in the submission screen used by the end user.

It has three possible configurations:

No

In this case the **Copy (cc)** attribute will not be available in the application, making it impossible to send emails with Copy(cc).

Yes

Neste caso o atributo **Cópia (cc)** estará disponível na aplicação para a configuração do desenvolvedor e na interface de envio de e-mail do usuário.

Hidden

In this case the **Copy (cc)** attribute will be available **only** for the developer configuration. The **Copy (cc)** field will not be displayed on the user's email sending interface screen, thus making it impossible to change the default values defined by the developer.

In this case, the email will always be sent with a copy to the emails configured by the developer.

Copy (CC)

Set the default value for the field in the application's end-user email sending interface.

Whether or not to inform a default value for the field is the choice of the developer, who can choose to activate the **empty attribute**, use a **global variable** or inform a **fixed text**.

> Whether using the variable or fixed text in the field, if you choose to inform more than two e-mails, they must be separated by **,** or **;**.

Empty attribute

When choosing not to inform a value, the developer forces the user to inform one or more destination e-mails.

Global variable

In this case, just add the variable to the value of the attribute containing the field's default email list.

fixed text

The developer can define one or more default emails directly in the attribute without using a variable.

It has blind carbon copy (bcc)

Defines the behavior of the **Copy (cc)** attribute in the submission screen used by the end user.

It has three possible configurations:

No

In this case the **Copy (cc)** attribute will not be available in the application, making it impossible to send emails with Copy(cc).

Yes

Neste caso o atributo **Cópia (cc)** estará disponível na aplicação para a configuração do desenvolvedor e na interface de envio de e-mail do usuário.

Hidden

In this case the **Copy (cc)** attribute will be available **only** for the developer configuration. The **Copy (cc)** field will not be displayed on the user's email sending interface screen, thus making it impossible to change the default values defined by the developer.

In this case, the email will always be sent with a copy to the emails configured by the developer.

Blind carbon copy (BCC)

Set the default value for the field in the application's end-user email sending interface.

Whether or not to inform a default value for the field is the choice of the developer, who can choose to activate the **empty attribute**, use a **global variable** or inform a **fixed text**.

> Whether using the variable or fixed text in the field, if you choose to inform more than two e-mails, they must be separated by **,** or **;**.

Empty attribute

When choosing not to inform a value, the developer forces the user to inform one or more destination e-mails.

Global variable

In this case, just add the variable to the value of the attribute containing the field's default email list.

fixed text

The developer can define one or more default emails directly in the attribute without using a variable.

Subject

Sets the default value for the field in the application's end-user email sending interface.

The developer can choose to define a default value using **global variable**, **lang variable**, informing a **fixed text** or can choose not to define a default text leaving the **attribute empty**.

> The default value defined by the developer can be changed by the application user.

Subject attribute configuration example

Subject field example in User interface

Body

Sets the default value for the field in the application’s end-user email sending interface.

The developer can choose to define a default value using **global variable**, **lang variable**, informing a **fixed text** or can choose not to define a default text leaving the **attribute empty**.

> The default value defined by the developer can be changed by the application user.

Subject attribute configuration example

Subject field example in User interface

Related Links

- Export Grid to Email
- Export Summary to Email

Related Video

- Export Setting

Grid Export PDF

Check the settings available in the grid’s PDF Export.

General settings

Export with password

This option defines the password used to open the exported file.

If this **option is disabled** the file can be opened by anyone who has access, if this option is enabled, a password must be defined that must be informed when trying to open the file.

> This option does not prevent the exported file from downloading.

Example of the password request screen:

Password

When enabling the option **Export with password** this attribute will be displayed for the definition of the password to access the file, which can contain between 1 and 255 alphanumeric characters.

> If no password is informed, the file will be generated without a password to access the file.

The file password can be set in two ways:

Entering the password directly

In this way, the exported files will always have the same access password.

Using a global variable

In this way the file password can be dynamically generated.

Select Columns

This attribute controls the display of the **Select Columns** tab in the generated application, which allows the user to define the fields that must be included in the export.

The fields positioned **to the right** in blue will be present in the export, whereas the fields **to the left** in red are outside the exported file.

PDF chart level selection

This attribute controls the display of the level selection option for the graph that will be generated in the PDF export.

When enabled, on the PDF export configuration screen, a list of fields that make up the current application break will be displayed.

Example of the Select level option

In the above image, the application has a grouping with three fields (levels).

- The **Level 1** would be the **Country**.
- The **Level 2** would be the **Creditlimit**.
- The **Level 3** would be the **Year of Birthdate**.

The selected level defines the amount of graphics that will be generated in the file, so when selecting level 3 of the grouping, as in the image, the PDF will be generated with 3 graphics, one for each level/field.

Selecting page break per level

This attribute enables the Fields for page break option, in the end-user export configuration screen, which defines the fields that will jump from page in the generated PDF.

Example:

Assuming that the user selects the **Year of Birthdate** field.

The PDF will be generated with each year of birth occupying its own page.

Year of Birthdate All records with the same year will be displayed on the page.

Open PDF Directly

This attribute defines whether or not the file access buttons are displayed after exporting.

If Disabled

The export screen will be displayed on the same application tab where the progress bar will be displayed with the file access buttons, allowing the user to choose between viewing the file, downloading it or returning to the source application.

The screen displayed will be as follows:

- **Preview** - Opens the generated file in a browser tab.
- **Download** - Download the generated file.
- **Back** - Returns to the source application.

  When enabling the option to generate the PDF directly, the **Display progress bar** attribute will be available.

Show progress bar

Only available when the attribute **Generate PDF Directly** is enabled, this flag controls the display of the progress bar when performing an export.

If enabled, a progress bar like this will be displayed where it is possible to follow the file generation process.

When not used, the progress bar will not be displayed when performing the export, in which case the user will not have any feedback on the progress of the process, which can be time consuming depending on the application used.

> We recommend using the d penablerogress bar when the amount of data processed is high or if using graphing.

Progress bar example

Default values settings and itens to end-user export interface

Configurable PDF

Defines whether or not to display the export configuration screen for the application’s end user.

When enabled, this feature enables the **PDF Configuration** screen, allowing the application user to configure the file in the best way, according to the options provided by the developer in the **PDF configurations available to the end user** field.

> The developer must set the default values for the fields available for user editing.

If disabled, the user will not be able to configure the PDF, the file will be generated according to the settings defined by the developer.

User setup screen example

Grid Export modules

Defines the query modules that will be added to the exported file.

If the **Grid export modules** field is added for user editing, the values defined by the developer will be the default value of the field in the user configuration interface.

> The **Graph** and **Summary** options are only available when there is one or more grouping configured in the application and when their respective modules are enabled in the menu Grid Modules{ :target='_blank'}.

Example

Grid export module configuration:

Example of the Export modules field in the application's user configuration interface:

Summary export modules

This option allows the developer to choose which modules will be available in the summary export. The options will be marked to be exported, however the user can change the modules at runtime.o.

If the field **Summary export modules** is added for user editing, the values defined by the developer will be the default value of the field in the user configuration interface.

> The **Graph** and **Summary** options are only available when there are one or more Breaks configured in the application and when their respective modules are enabled in the Grid Modules{ link: target='_blank'}.

Example

Summary Export Modules Configuration:

On execution, the Summary option will be disabled:

Print Type

Allows you to set the print mode of the file (Economic, Color).

Economical: Black and white printing Color: Color Printing

PDF Format

This option allows you to define the format of the report pages (letter, A4, etc).

PDF Orientation

Allows you to define whether the print will be in Portrait or Landscape orientation.

Generate bookmarks

This option creates bookmarks, which facilitate navigation between pdf records.

If the option to generate bookmarks is enabled when exporting the PDF, bookmarks will be generated in the upper left corner of the screen, as shown in the image below:

> **Bookmarkers** will only be generated if a group by is defined in the application.

Information that appears if the bookmarkers option is enabled:

- Breakage information
- Information about the query totalization
- Information about the query summary

Display header in all pages

Allows the header to be displayed even if it is not available in the query.

> This option, by default, is disabled in the pdf configuration interface, causing the header to be displayed. To enable this option in the pdf configuration interface, uncheck the **Display title on all pages** option box.

Display the title on all pages

This option will display the column headings on all pages of the PDF. If this option is unchecked, the title will only be displayed on the first page.

Display title below each Group By

When this option is enabled, it will display a box already checked by default in the PDF configuration, where you can uncheck it if desired. If the box is checked, it will display the column headings in the PDF in all available groupings.

Compressed PDF

It generates the .PDF file compressed in .zip format, allowing the reduction of the generated file.

> This option makes PDF generation a little more time consuming.

PDF Settings

Records per Page

Allows you to define the number of lines per page of the query that will be displayed in the file.

Records per page in the summary

Allows you to define the number of lines per page of the summary that will be displayed in the file.

Complete lines to footer

Complete with (empty) lines to the footer.

Page numbering format

Page Numbering Format whether it will be simple(1,2,3,4,5,...), Full(1/n, 2/n, 3/n,...) or not use numbering.

Height of page numbering

Allows you to align the numbering vertically, at the top or bottom of the page.

Horizontal position of page numbering

Allows you to horizontally align the page numbering (Left, Center, Right).

Margin

The value in millimeters to be applied to the PDF margins must be filled in (Top, Bottom, Right, Left). Mais sobre o texto original

> The filling of the value in millimeters applies to the margin (up, down, right and left).

print background

Allows you to print background on the PDF file.

JS execution time

Maximum time (in seconds) the server waits when running JS.

Timeout for chart’s image creation

Allows you to define whether the application’s graphics will be displayed in the PDF.

Word

General Settings

Export with password

This option defines the password used to open the exported file.

If this **option is disabled** the file can be opened by anyone who has access, if this option is enabled, a password must be defined that must be informed when trying to open the file.

> This option does not prevent the exported file from downloading.

Example of the password request screen:

Password

When enabling the option **Export with password** this attribute will be displayed for the definition of the password to access the file, which can contain between 1 and 255 alphanumeric characters.

> If no password is informed, the file will be generated without a password to access the file.

The file password can be set in two ways:

Entering the password directly

In this way, the exported files will always have the same access password.

Using a global variable

In this way the file password can be dynamically generated.

Select Columns

This attribute controls the display of the **Select Columns** tab in the generated application, which allows the user to define the fields that must be included in the export.

The fields positioned **to the right** in blue will be present in the export, whereas the fields **to the left** in red are outside the exported file.

Open WORD Directly

This attribute defines whether or not the file access buttons are displayed after exporting.

If Disabled

The export screen will be displayed on the same application tab where the progress bar will be displayed with the file access buttons, allowing the user to choose between viewing the file, downloading it or returning to the source application.

The screen displayed will be as follows:

- **Preview** - Opens the generated file in a browser tab.
- **Download** - Download the generated file.

- **Back** - Returns to the source application.

Default values settings and itens to end-user export interface

Word configurable

Defines whether or not to display the export configuration screen at runtime.

If Enabled

A screen with configuration options for the exported file will be displayed, allowing the end user to configure the export the best way.

Configuration options must be defined by the developer in the **WORD settings available to the end user** field.

> In this case, the settings defined by the developer will serve as default values for the user configuration screen.

Setup screen example

If Disabled

In this case, the settings defined by the developer will be applied to the file generated in the export.

Grid Export modules

This option allows the developer to choose which modules will be available when exporting the grid. The options will be marked to be exported, however the user can change the modules at runtime.

Example:

Disabling the Summary option:

On execution, the Summary option will be disabled:

Summary export modules

This option allows the developer to choose which modules will be available in the summary export. The options will be checked to be exported, however the user can change the modules at runtime.

Exemple:

No option disabled in Summary export:

On execution, the Summary option will be disabled:

Word settings

Lines per page in grid

Allows you to define the number of lines per page of the grid that will be displayed in the file.

Lines per page in sumary

Allows you to define the number of lines per page of the summary that will be displayed in the file.

CSV

General Settings

Export with password

This option defines the password used to open the exported file.

If this **option is disabled** the file can be opened by anyone who has access, if this option is enabled, a password must be defined that must be informed when trying to open the file.

> This option does not prevent the exported file from downloading.

Example of the password request screen:

Password

When enabling the option **Export with password** this attribute will be displayed for the definition of the password to access the file, which can contain between 1 and 255 alphanumeric characters.

> If no password is informed, the file will be generated without a password to access the file.

The file password can be set in two ways:

Entering the password directly

In this way, the exported files will always have the same access password.

Using a global variable

In this way the file password can be dynamically generated.

Select Columns

This attribute controls the display of the **Select Columns** tab in the generated application, which allows the user to define the fields that must be included in the export.

The fields positioned **to the right** in blue will be present in the export, whereas the fields **to the left** in red are outside the exported file.

Open CSV Directly

This attribute defines whether or not the file access buttons are displayed after exporting.

If Disabled

The export screen will be displayed on the same application tab where the progress bar will be displayed with the file access buttons, allowing the user to choose between viewing the file, downloading it or returning to the source application.

The screen displayed will be as follows:

- **Preview** - Opens the generated file in a browser tab.
- **Download** - Download the generated file.

- **Back** - Returns to the source application.

Default values settings and itens to end-user export interface

Configurable CSV

Sets whether or not to display it on the export configuration screen at runtime

If Enabled

A screen with configuration options for the exported file will be displayed, allowing the end user to configure the export the best way.

Configuration options must be defined by the developer in the **WORD settings available to the end user** field.

> In this case, the settings defined by the developer will serve as default values for the user configuration screen.

Setup screen example

If Disabled

In this case, the settings defined by the developer will be applied to the file generated in the export.

Grid Export modules

This option allows the developer to choose which modules will be available when exporting the grid. The options will be marked to be exported, however the user can change the modules at runtime.

Example:

Disabling the Summary option:

On execution, the Summary option will be disabled:

Summary export modules

This option allows the developer to choose which modules will be available in the summary export. The options will be checked to be exported, however the user can change the modules at runtime.

Exemple:

No option disabled in Summary export:

On execution, the Summary option will be disabled:

Line separator

Allows you to set the line separator character.

Column separator

Allows you to define the separator character of columns and records.

Text delimiter

Allows you to define the character used to delimit the text of the columns.

Add label

Allows you to define whether the column label will be added to the file.

Excel

General Settings

Export with password

This option defines the password used to open the exported file.

If this **option is disabled** the file can be opened by anyone who has access, if this option is enabled, a password must be defined that must be informed when trying to open the file.

> This option does not prevent the exported file from downloading.

Example of the password request screen:

Password

When enabling the option **Export with password** this attribute will be displayed for the definition of the password to access the file, which can contain between 1 and 255 alphanumeric characters.

> If no password is informed, the file will be generated without a password to access the file.

The file password can be set in two ways:

Entering the password directly

In this way, the exported files will always have the same access password.

Using a global variable

In this way the file password can be dynamically generated.

Select Columns

This attribute controls the display of the **Select Columns** tab in the generated application, which allows the user to define the fields that must be included in the export.

The fields positioned **to the right** in blue will be present in the export, whereas the fields **to the left** in red are outside the exported file.

Open Excel Directly

This attribute defines whether or not the file access buttons are displayed after exporting.

If Disabled

The export screen will be displayed on the same application tab where the progress bar will be displayed with the file access buttons, allowing the user to choose between viewing the file, downloading it or returning to the source application.

The screen displayed will be as follows:

- **Preview** - Opens the generated file in a browser tab.
- **Download** - Download the generated file.

- **Back** - Returns to the source application.

Default values settings and itens to end-user export interface

Excel Configurable

Defines whether or not to display the export configuration screen at runtime.

If Enabled

A screen with configuration options for the exported file will be displayed, allowing the end user to configure the export the best way.

The configuration options must be defined by the developer in the field **Excel settings available to the end user.**.

> In this case, the settings defined by the developer will serve as default values for the user configuration screen.

Setup screen example

If Disabled

In this case, the settings defined by the developer will be applied to the file generated in the export.

Grid Export modules

This option allows the developer to choose which modules will be available when exporting the grid. The options will be marked to be exported, however the user can change the modules at runtime.

Example:

Disabling the Summary option:

On execution, the Summary option will be disabled:

Summary export modules

This option allows the developer to choose which modules will be available in the summary export. The options will be marked to be exported, however the user can change the modules at runtime.

Example:

No option disabled in Summary export:

On execution, the Summary option will be disabled:

Format

Allows you to define the format of the generated document (xls or xls).

Export with totals

Allows the display of the total when exporting to Excel.

XML

General settings

![Configuração de exportação Word da consulta][img_conf_geral_exportar_xml]

Export with password

This option defines the password used to open the exported file.

If this **option is disabled** the file can be opened by anyone who has access, if this option is enabled, a password must be defined that must be informed when trying to open the file.

> This option does not prevent the exported file from downloading.

Example of the password request screen:

Password

When enabling the option **Export with password** this attribute will be displayed for the definition of the password to access the file, which can contain between 1 and 255 alphanumeric characters.

> If no password is informed, the file will be generated without a password to access the file.

The file password can be set in two ways:

Entering the password directly

In this way, the exported files will always have the same access password.

Using a global variable

In this way the file password can be dynamically generated.

Select Columns

This attribute controls the display of the **Select Columns** tab in the generated application, which allows the user to define the fields that must be included in the export.

The fields positioned **to the right** in blue will be present in the export, whereas the fields **to the left** in red are outside the exported file.

Open XML Directly

This attribute defines whether or not the file access buttons are displayed after exporting.

If Disabled

The export screen will be displayed on the same application tab where the progress bar will be displayed with the file access buttons, allowing the user to choose between viewing the file, downloading it or returning to the source application.

The screen displayed will be as follows:

- **Preview** - Opens the generated file in a browser tab.
- **Download** - Download the generated file.

- **Back** - Returns to the source application.

Default values settings and itens to end-user export interface

Configurable XML

Defines whether or not to display the export configuration screen at runtime.

If Enabled

A screen with configuration options for the exported file will be displayed, allowing the end user to configure the export the best way.

The configuration options must be defined by the developer in the field **Excel settings available to the end user.**.

> In this case, the settings defined by the developer will serve as default values for the user configuration screen.

Setup screen example

If Disabled

In this case, the settings defined by the developer will be applied to the file generated in the export.

Grid Export modules

This option allows the developer to choose which modules will be available when exporting the grid.

The options will be marked to be exported, however the user can change the modules at runtime.

Example:

Disabling the Summary option:

On execution, the Summary option will be disabled:

Summary export modules

This option allows the developer to choose which modules will be available in the summary export. The options will be marked to be exported, however the user can change the modules at runtime.

Example:

No option disabled in Summary export:

On execution, the Summary option will be disabled:

Format

Allows you to define the format of the generated document (xls or xls).

Export with totals

Allows the display of the total when exporting to Excel. [img_conf_geral_exportar_xml]:

/assets/images/docs/app/comum/exportacao/conf_valores_padrao_usr/xml/conf_geral_exportar_xml.png

JSON

General settings

Export with password

This option defines the password used to open the exported file.

If this **option is disabled** the file can be opened by anyone who has access, if this option is enabled, a password must be defined that must be informed when trying to open the file.

> This option does not prevent the exported file from downloading.

Example of the password request screen:

Password

When enabling the option **Export with password** this attribute will be displayed for the definition of the password to access the file, which can contain between 1 and 255 alphanumeric characters.

> If no password is informed, the file will be generated without a password to access the file.

The file password can be set in two ways:

Entering the password directly

In this way, the exported files will always have the same access password.

Using a global variable

In this way the file password can be dynamically generated.

Select Columns

This attribute controls the display of the **Select Columns** tab in the generated application, which allows the user to define the fields that must be included in the export.

The fields positioned **to the right** in blue will be present in the export, whereas the fields **to the left** in red are outside the exported file.

Generate JSON directly

This attribute defines whether or not the file access buttons are displayed after exporting.

If Disabled

The export screen will be displayed on the same application tab where the progress bar will be displayed with the file access buttons, allowing the user to choose between viewing the file, downloading it or returning to the source application.

The screen displayed will be as follows:

- **Preview** - Opens the generated file in a browser tab.
- **Download** - Download the generated file.

- **Back** - Returns to the source application.

Default values settings and itens to end-user export interface

Configurable JSON

Defines whether or not to display the export configuration screen at runtime.

If Enabled

A screen with configuration options for the exported file will be displayed, allowing the end user to configure the export the best way.

Configuration options must be defined by the developer in the **JSON settings available to the end user.** field.

> In this case, the settings defined by the developer will serve as the default values for the user configuration screen.

Setup screen example

If Disabled

In this case, the settings defined by the developer will be applied to the file generated in the export

Grid Export modules

This option allows the developer to choose which modules will be available when exporting the grid.

The options will be marked to be exported, however the user can change the modules at runtime.

Example:

Disabling the Summary option:

On execution, the Summary option will be disabled:

Módulos de exportação do resumo

Summary export modules This option allows the developer to choose which modules will be available in the summary export.

The options will be marked to be exported, however the user can change the modules at runtime.

Example:

No option disabled in Summary export:

On execution, the Summary option will be disabled:

Use Label

Allows the label used in the application field to be sent in the JSON attribute.

Formatted Value

Allows you to display the formatted numeric value or the displayed value in the database.

Impressão HTML

General settings

Export with password

This option defines the password used to open the exported file.

If this **option is disabled** the file can be opened by anyone who has access, if this option is enabled, a password must be defined that must be informed when trying to open the file.

> This option does not prevent the exported file from downloading.

Example of the password request screen:

Password

When enabling the option **Export with password** this attribute will be displayed for the definition of the password to access the file, which can contain between 1 and 255 alphanumeric characters.

> If no password is informed, the file will be generated without a password to access the file.

The file password can be set in two ways:

Entering the password directly

In this way, the exported files will always have the same access password.

Using a global variable

In this way the file password can be dynamically generated.

Select Columns

This attribute controls the display of the **Select Columns** tab in the generated application, which allows the user to define the fields that must be included in the export.

The fields positioned **to the right** in blue will be present in the export, whereas the fields **to the left** in red are outside the exported file.

Open directly

This attribute defines whether or not the file access buttons are displayed after exporting.

If Disabled

The export screen will be displayed on the same application tab where the progress bar will be displayed with the file access buttons, allowing the user to choose between viewing the file, downloading it or returning to the source application.

The screen displayed will be as follows:

- **Preview** - Opens the generated file in a browser tab.
- **Download** - Download the generated file.

- **Back** - Returns to the source application.

Default values settings and itens to end-user export interface

Configurable Print HTML

Defines whether or not to display the export configuration screen at runtime.

If Enabled

A screen with configuration options for the exported file will be displayed, allowing the end user to configure the export the way the best way.

The configuration options must be defined by the developer in the field **HTML print settings available to the end user**.

> In this case, the settings defined by the developer will serve as the default values for the user configuration screen.

Setup screen example

If Disabled

In this case, the settings defined by the developer will be applied to the file generated in the export

Grid Export modules

This option allows the developer to choose which modules will be available when exporting the grid.

The options will be marked to be exported, however the user can change the modules at runtime.

Defines the modules of the grid that will be added to the exported file.

> The **Graph** and **Summary** options are only available when there are one or more Group By configured and when their modules are enabled in the menu Grid Modules.

Exemplo:

Disabling the Summary option:

On execution, the Summary option will be disabled:

Summary export modules

This option allows the developer to choose which modules will be available in the summary export. The options will be marked to be exported, however the user can change the modules at runtime.

Example:

Disabling the Summary option:

On execution, the Summary option will be disabled:

Print mode

Allows you to define the print content of the file (Both, Current page, Full report).

Print Type

Permite definir o tipo de impressão do arquivo. (Ambos, Preto e Branco , Colorido).

Print settings

Records per page

Allows you to define the number of lines per page of the query that will be displayed in the file.

Records per page in the summary

Allows you to define the number of lines per page of the summary that will be displayed in the file.

Print background

Allows you to define whether the background will be displayed when printing.

RTF

General settings

Select Columns

This attribute controls the display of the **Select Columns** tab in the generated application, which allows the user to define the fields that must be included in the export.

The fields positioned **to the right** in blue will be present in the export, whereas the fields **to the left** in red are outside the exported file.

Open RTF Directly

This attribute defines whether or not the file access buttons are displayed after exporting.

If Disabled

The export screen will be displayed on the same application tab where the progress bar will be displayed with the file access buttons, allowing the user to choose between viewing the file, downloading it or returning to the source application.

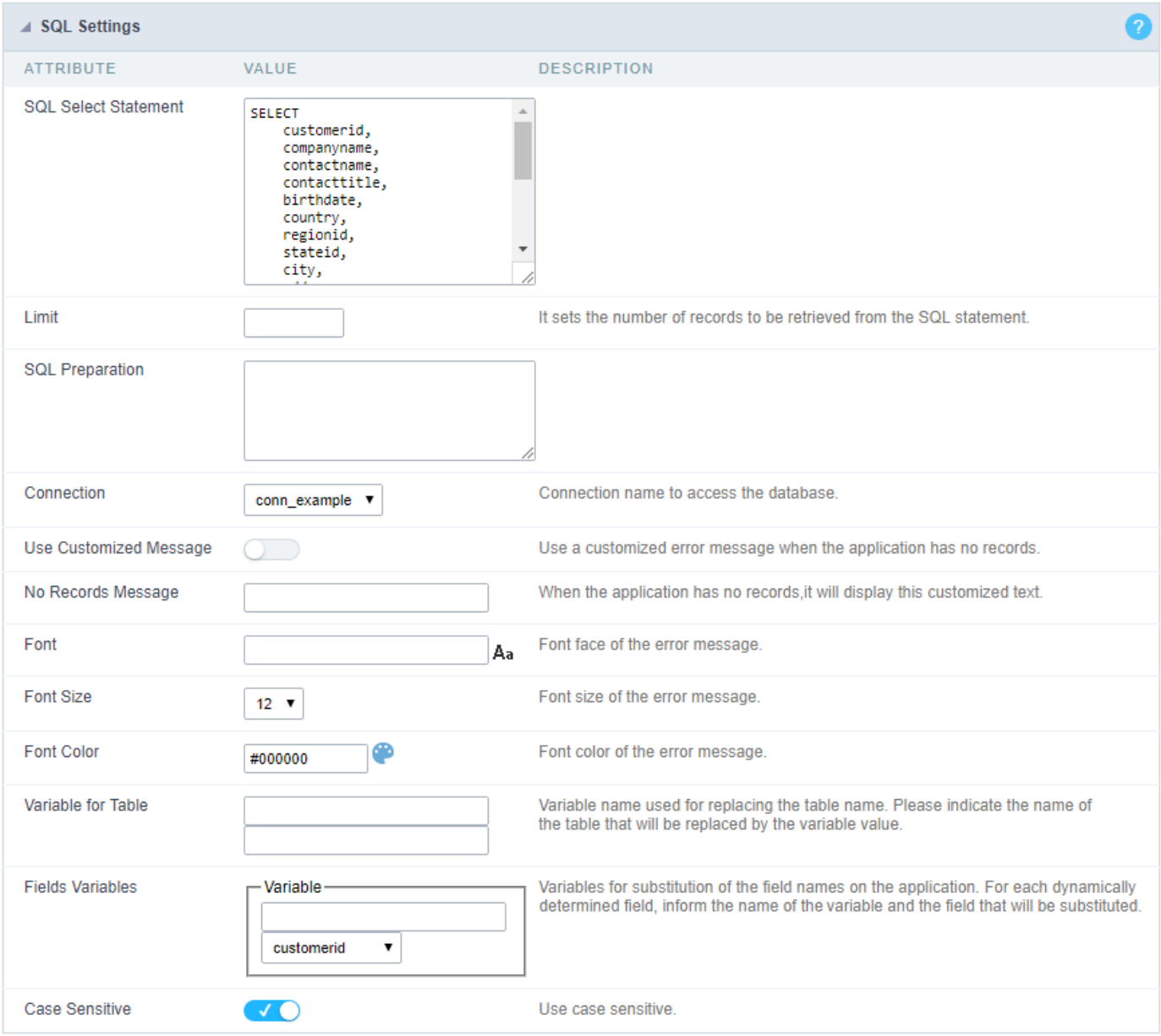
The screen displayed will be as follows:

- **Preview** - Opens the generated file in a browser tab.
- **Download** - Download the generated file.
- **Back** - Returns to the source application.

Grid SQL Settings

SQL Settings

This interface allows configuring the related database settings, such as the SQL statement, the used database connection, case sensitive, and others.



Grid SQL configuration

SQL Select Statement

It allows you to define the primary SQL of the application. You can edit this SQL to add or delete fields.

Limit

Lets you limit the display in the number of records retrieved by SQL query.

SQL Preparation

You can enter SQL commands or procedure names to execute them before the primary SQL of the application.

Connection

It allows defining the database connection of the application. You can change the connection to another one that has the same table.

Use Customized Message

Lets you define to display the “no records” message or not.

No Records Message

Lets you set the message when the application has no records.

Font

This option is available when using the “Use Customized Message” option. It lets you set the font for the message.

Font Size

This option is available when using the “Use Customized Message” option. It lets you to set the font size.

Font Color

This option is available when using the “Use Customized Message” option. It lets you to set the font color.

Variable for Table

It allows to use a variable to change a part of the string containing the table name.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the table you want to replace (replace from).

Fields Variables

It allows to use a variable to change a part of the string containing the field name.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the field you want to replace (replace from).

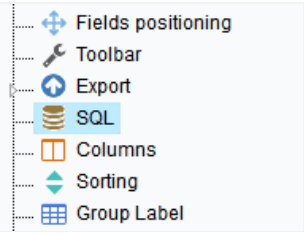
Case sensitive

It defines if the database connection uses case sensitive or not.

Creating Parameters for Applications Links

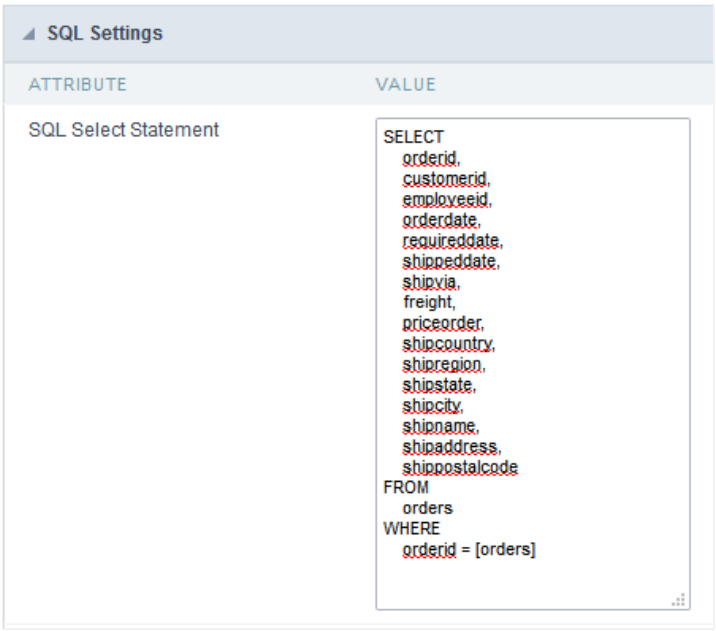
Parameters are used to pass values between applications. When the target application is a query, we recommend adding a global variable in the WHERE clause to filter the records that meet a specific condition.

To configure the parameter, in the destination application, access the SQL option in the left side menu.



We must edit the SQL of the grid application, adding the WHERE clause and the field that receive the value through a global

variable(as shown in the image below).



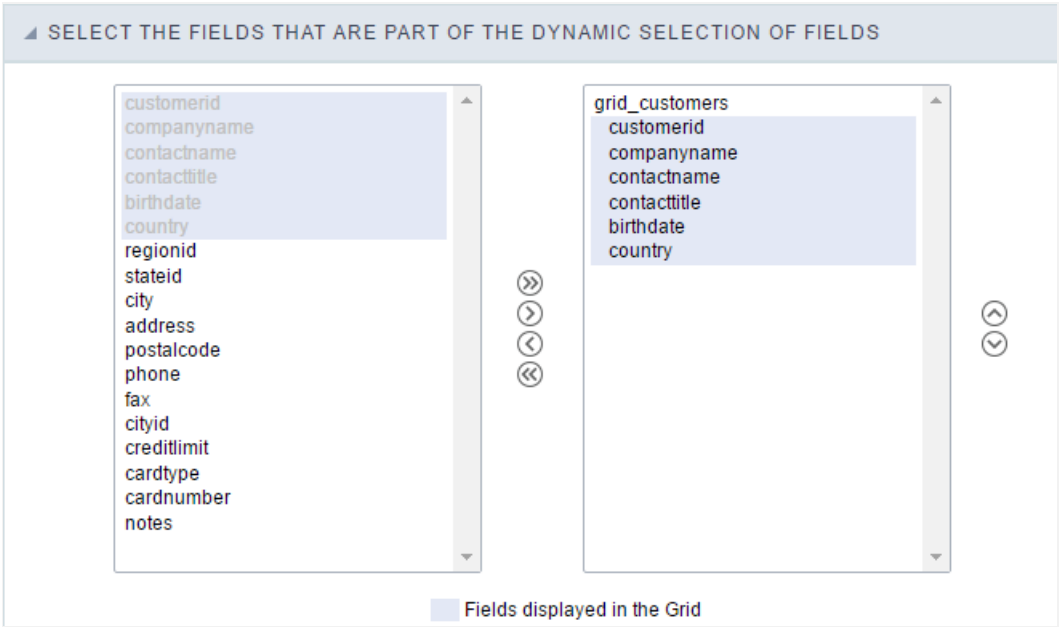
after edition, we must generate the source code of the application clicking on the button -

- or using the hotkey **F8**.

Related Videos ▷

Columns

You can display or hide fields dynamically inside the Grid application using the button “Columns”. This option is used to select the fields/columns that will be part of this dynamic fields selection.



Grid columns’ settings

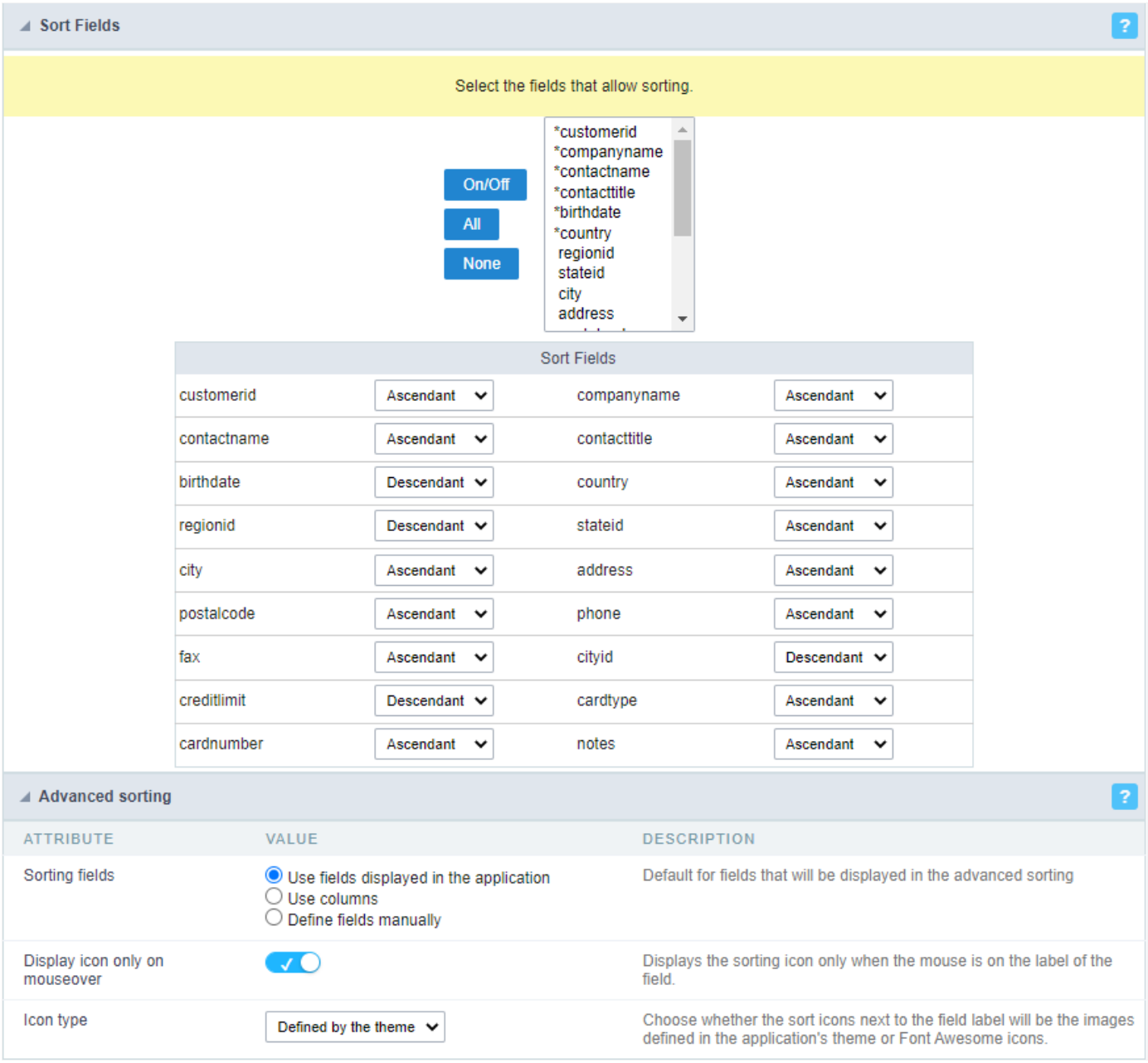
The box in the left represents the fields that are NOT being displayed within the dynamic fields selection.

The box in the right represents the fields that are being displayed within the dynamic fields selection.

Sorting Settings

They define the fields available for sorting options in the runtime application and their behavior.

To configure the application's initial ordering, it is necessary to add a **order by** in the query query in the **SQL** menu.

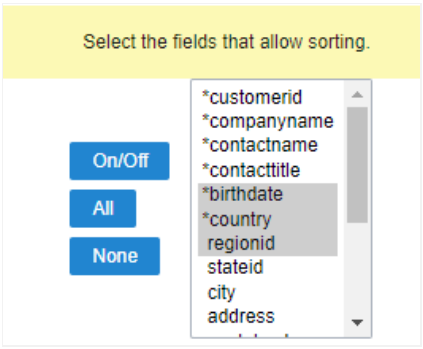


The settings in the **Fields with Sorting** block refer to the application fields, whereas the **Advanced Sorting** options refer to the button on the toolbar.

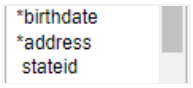
Fields with Ordering

This definition is made using the **On/Off** button which, when activated, inverts the current definition of the selected fields.

By default, the first six fields of the application are enabled to be available for sorting

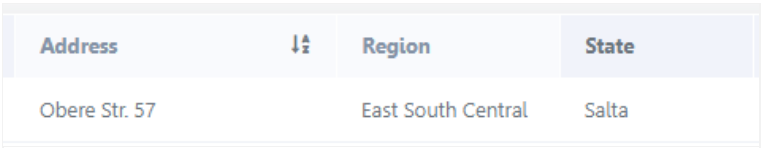


The active fields for sorting in the application are represented by the asterisk next to the name. In the example below the **birthdate** and **address** fields will be available for sorting while the **stateid** field will not be.

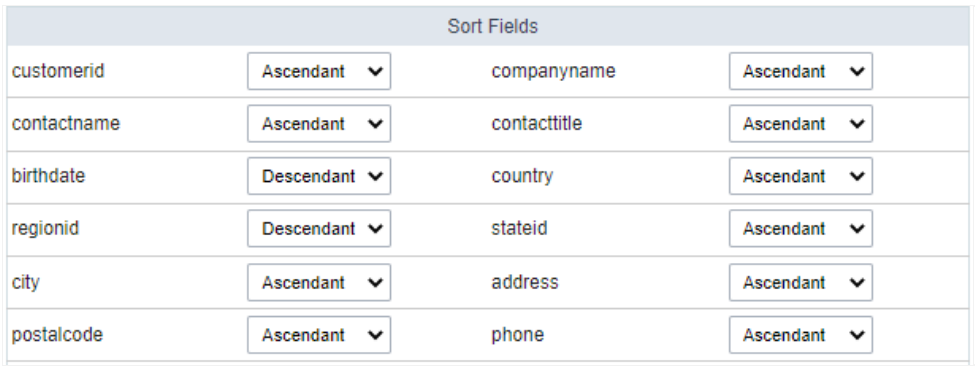


In the application, the fields available for sorting can be identified by the sort icon displayed on the hover (when passing the mouse over) of the columns.

So that the icons are always displayed we must change the option **Display icon only on mouseover** in the advanced settings below.



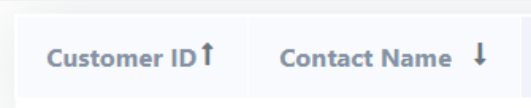
Classify Fields



In this option all fields of the application will be displayed, it defines the initial behavior of the field when clicked to perform the ordination.

For example, the **customerid** field is selected to perform an **Ascending** sort on the first click, whereas the **contacname** field is set to start **Descending** sorting.

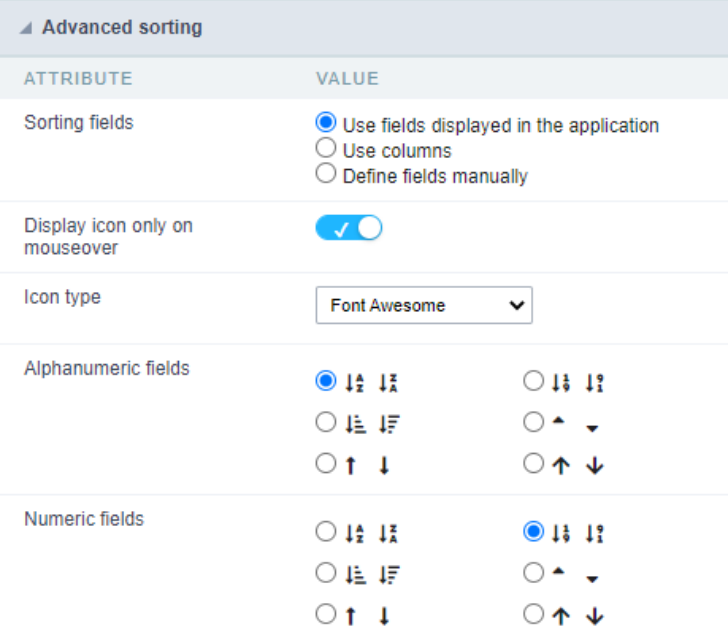
This configuration can also be identified in the application according to the icon displayed on the hover of the field



This In this way, when clicking on the **contacname** field, a descending order will be performed at first.

Advanced Sorting

Sorting These options define the fields available in the sort button on the toolbar, as well as the behavior and icons of the fields available for sorting in the application.



Sorting fields

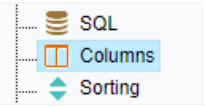
In this attribute we define the fields available for sorting in the **sort** field of the toolbar.

Use fields displayed in the application

The fields defined for display in the application in the **field placement** menu will be available for sorting.

Use columns

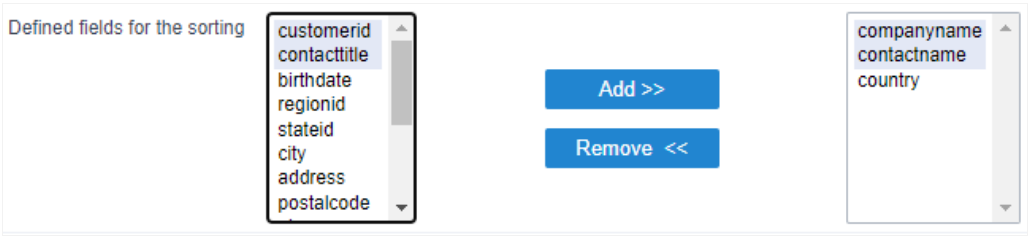
In this option, the **order** button inherits all the fields selected to integrate the columns button. These fields are defined in the **Columns** menu.



Manually define fields

When selecting this option a new attribute, **Fields defined for sorting** will be displayed to define the available fields in the sort button.

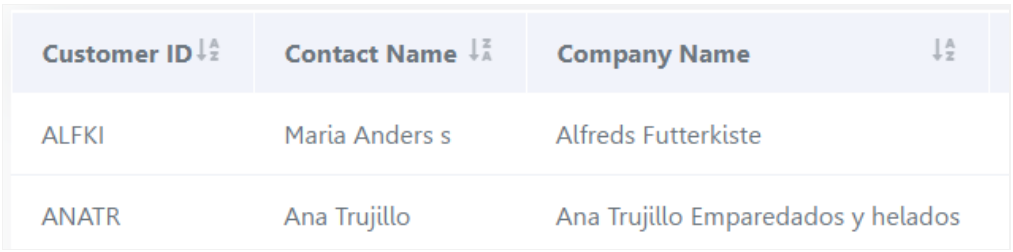
The highlighted fields, for example **customerid** and **address**, are the fields that are currently being displayed in the application.



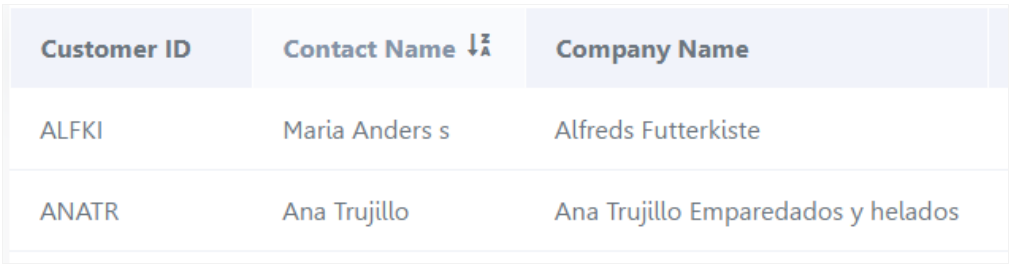
Display icon only on mouseover

This attribute defines if the sort icon will always be visible or if it will only be available on mouseover of the field.

Show icon on mouseover only option disabled



Display icon on mouseover only option enabled



Icon Type

Defines if the icon will be inherited by the theme when selecting **Defined by theme** or if it will use **font Awesome** which enables two other attributes: **Alphanumeric Fields** and **Numeric Fields**.

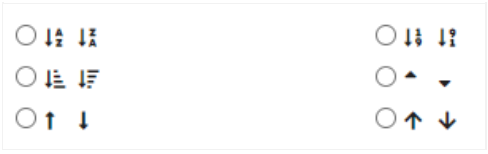
In the project theme it is not possible to select the **font Awesome** option to define the sorting icons.

Alphanumeric Fields

Defines the sort icon for all alphanumeric fields in the application.

Numerical Fields

Defines the sort icon for all numeric fields in the application.

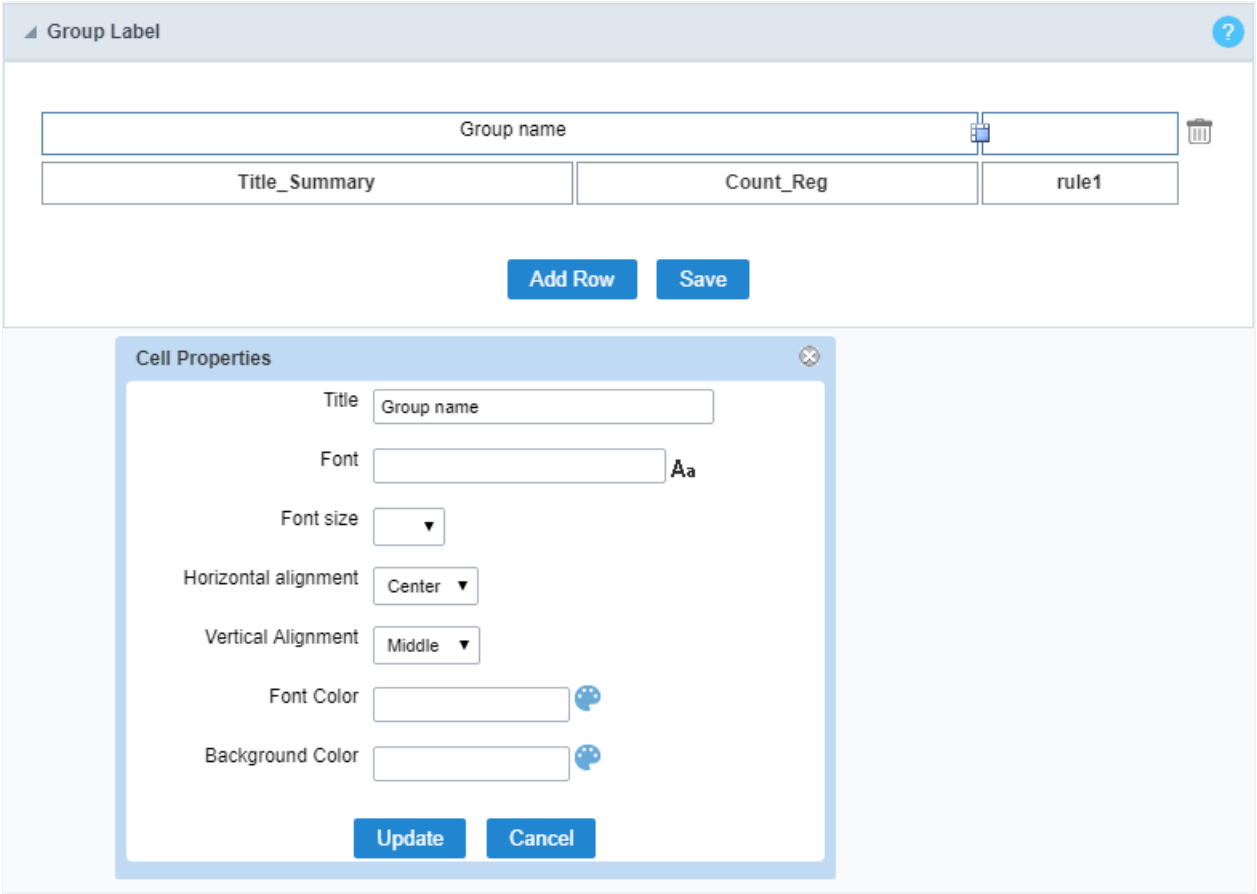


Related Video ▷

Grid Group Label

With this option, you can insert one or more titles for the columns in the application, allowing to modify its size, color, font, and position.

The Group Label doesn't work with a dynamic display of fields.



Editing Group Label cell properties

Title

It allows you to define a Title for the Group Label.

Font

Set the font family for the group label title.

Font Size

Set the font size for the group label title.

Horizontal Alignment

Define the horizontal alignment of the group label title. Left, Center, or Right.

Vertical Alignment

Define the vertical alignment of the group label title. Middle, Bottom, or Top.

Font Color

Set the font color for the group label title.

Background Color

Define a background color for the group label title.

Configuring Security in Grid

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application.

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Configuring Security Cookies in Grid

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Configuring Security Headers in Grid

Desabilitar o Auditor XSS

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

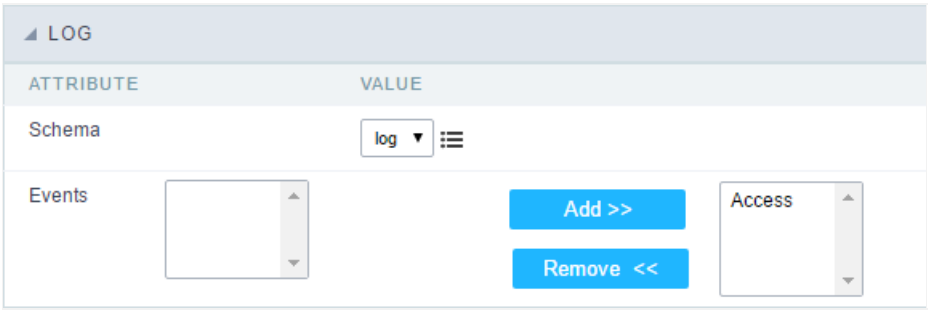
Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Grid Log Configurations

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Related Videos ▷

Gantt Chart

The **Gantt chart** is a visual representation of activities over time, allowing you to track deadlines, duration, and overlapping tasks.

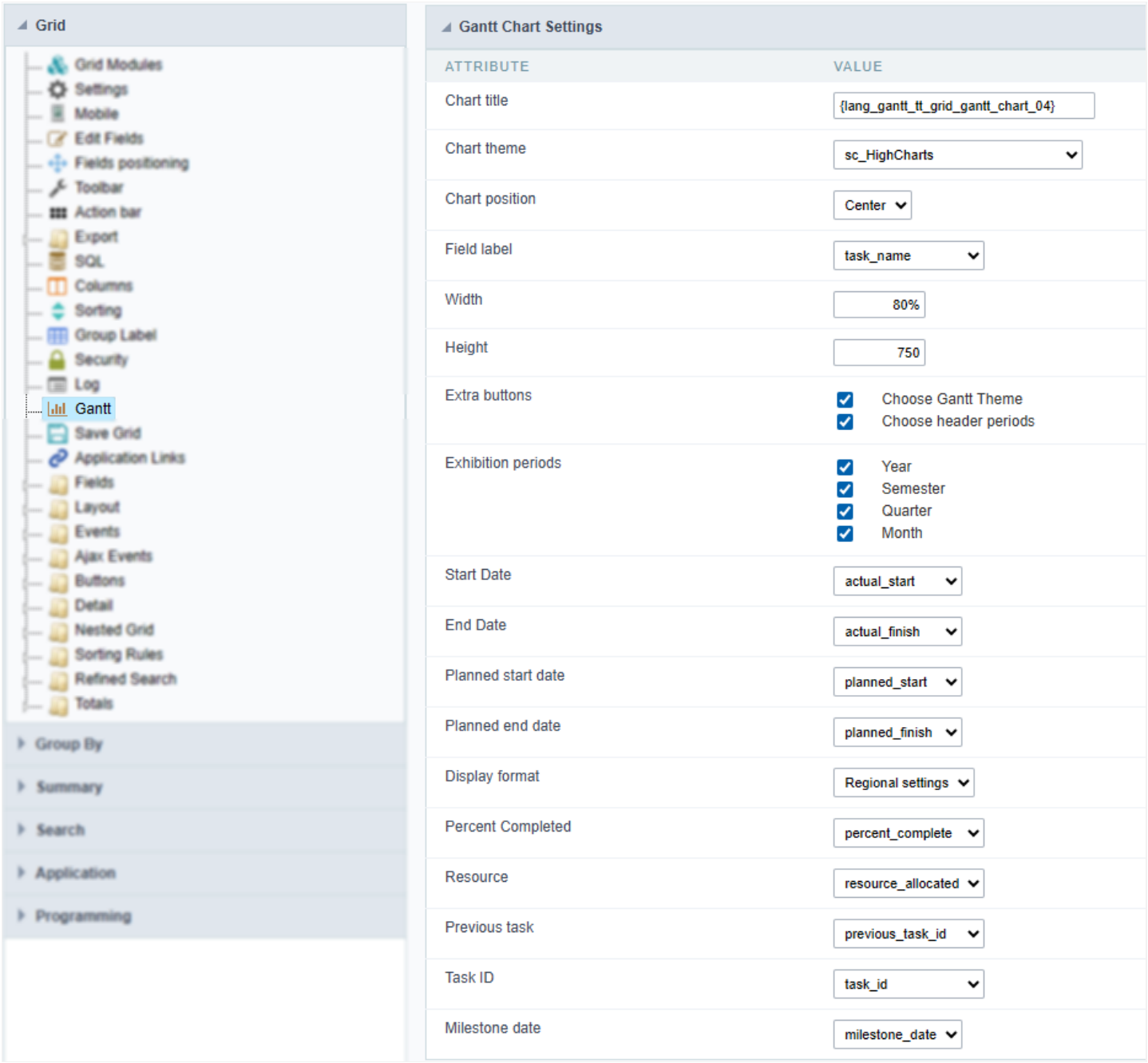


Chart Title

Defines the title of the Gantt chart that will be displayed in the application.

The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.

See more details about langs and about our Data dictionary.

Example of the chart title displayed in the application.

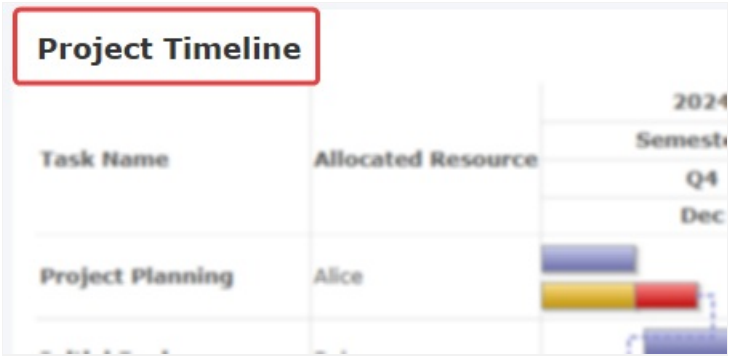


Chart Theme

This attribute defines the theme used for the Gantt chart, working independently from the application's general theme.

Available options:

- **Inherit application theme**: Applies the chart theme defined in the application's general theme settings. Themes can be edited via *Layout > Themes for applications*.
- **Theme list**: Displays the themes created under *Layout > Chart Themes*.

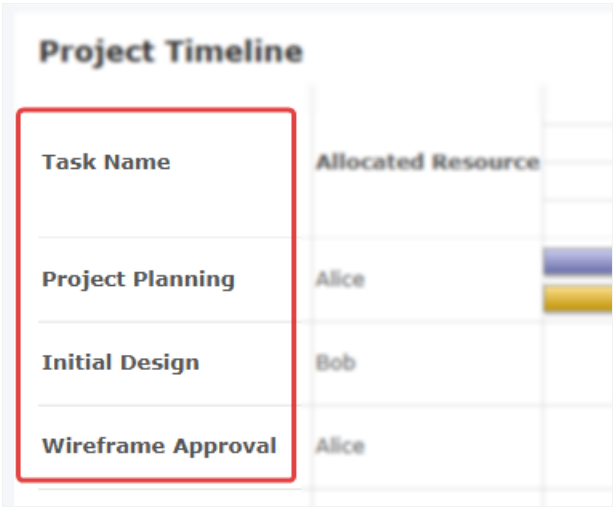
Chart Position

Defines the position of the application on the screen when the Gantt chart is displayed. The application can be aligned to the **Left**, **Center**, or **Right**.

Label Field

Defines the field that will be displayed in the main column of the Gantt chart. This attribute is required for the chart to be displayed.

Example of a Gantt chart with a task list, where the attribute is set to the task name field.



Width

Defines the available width for displaying the Gantt chart.

This field accepts values in **pixels** (e.g., `800px`) or **percentage** (e.g., `100%`), allowing you to control how the chart adapts to the available screen space.

Height

Allows you to set the height of the Gantt chart using fixed values in **pixels** (e.g., `600px`).

Unlike width, this field **does not accept percentage values**. The height must be defined with a fixed size to ensure proper display of the entire chart.

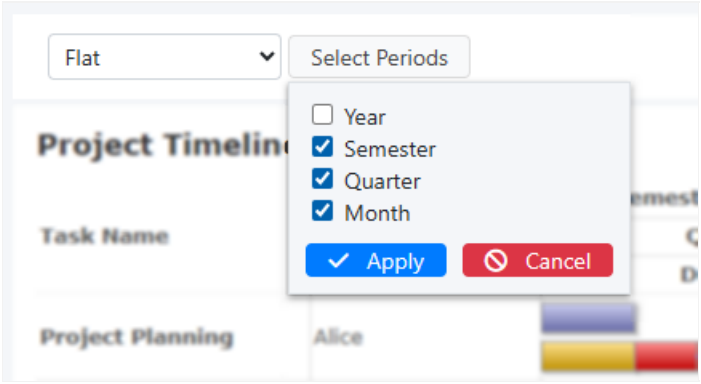
Extra Buttons

Allows you to configure buttons displayed in the Gantt chart's toolbar.

Available options

- **Choose Gantt theme**: When enabled, displays a combo box with the list of available chart themes.
- **Select periods in the title bar**: Enables a button in the chart toolbar that allows users to change the period view configured in the application.

Example of the buttons in the interface.

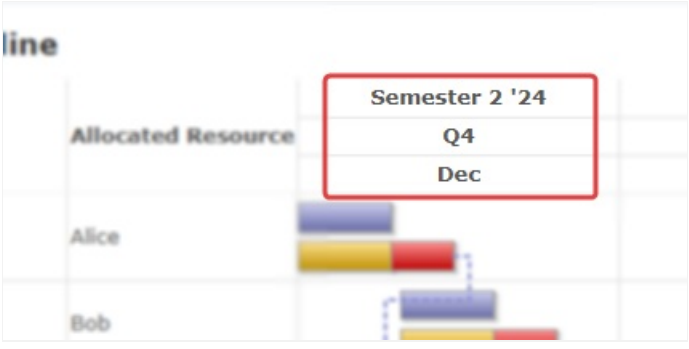


Display Periods

Use this attribute to define the time intervals for data display in the chart, which can be: **Year**, **Semester**, **Quarter**, and **Month**.

This attribute also defines the available values in the **Set Periods** button.

Example of the chart with all periods selected.



Start Date

Indicates the actual start date of the activity in the Gantt chart.

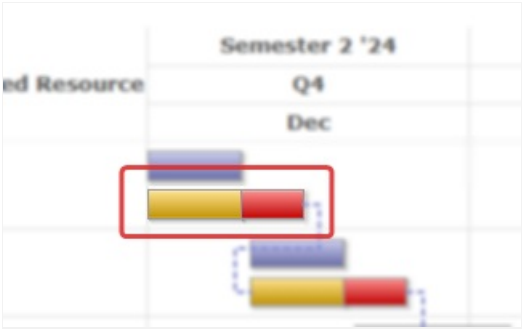
This attribute must be assigned a date field that stores this information, essential for displaying the execution bar in the chart.

End Date

This attribute indicates the actual completion date of the activity.

Along with the **Start Date** field, it is responsible for rendering the execution bar that represents the progress of the activity.

Example of the execution bar in the Gantt chart.



Planned Start Date

Together with the **Planned End Date** attribute, it is responsible for displaying the planning bar in the Gantt chart.

This field should be mapped to a date field that contains the planned start date for the activity.

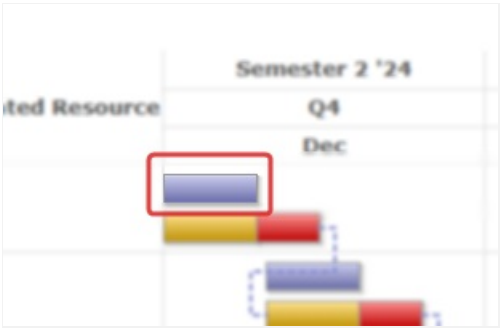
This field is optional. However, if left blank, the planning bar will not be displayed in the chart.

Planned End Date

Responsible for rendering the planning bar in the Gantt chart together with the **Planned Start Date** field.

This field must be mapped to a date field that stores the planned end date of the activity.

Example of the planning bar in the Gantt chart.



This field is optional. However, if left blank, the planning bar will not be displayed in the chart.

Display Format

Allows you to define the date format displayed in the Gantt chart.

Available options:

- **Use Regional Settings**: This is the default option. It allows the date format to adapt according to the selected language. You can customize regional settings in your project via Locales > Regional Settings.
- **dd/mm/yyyy**: displays the date in day/month/year format (e.g., 25/10/2025).
- **mm/dd/yyyy**: displays the date in month/day/year format (e.g., 10/25/2025), mainly used in English (US) systems.

Percent Complete

Defines the activity progress as a percentage, indicating the status of the task.

This field must contain a numeric value between 0 and 100, representing the completion percentage of the activity. This value is used to display a progress bar within the Gantt chart.

This field is **ignored** when both the **Planned Start Date** and **Planned End Date** fields are filled. In such cases, the chart uses the planned start and end dates to calculate progress and delay.

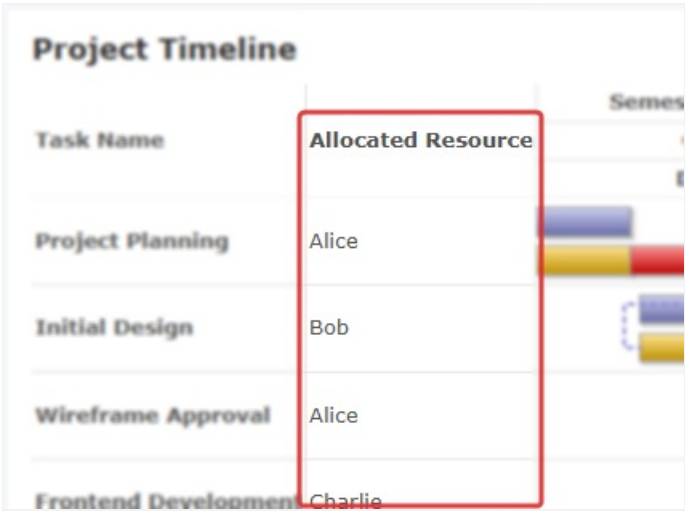
Resource

Defines the resource used for each activity.

This field can represent a person, team, or any other element assigned to perform the activity.

When configured, this field adds a second column to the Gantt chart displaying the assigned resources.

Example of the resource column in the chart:



Predecessor Task

Defines the dependency between activities in the Gantt chart.

The “Predecessor Task” should contain the identifier (ID) of the activity that must be completed before the current one starts. This relationship creates a visual dependency line between the chart bars.

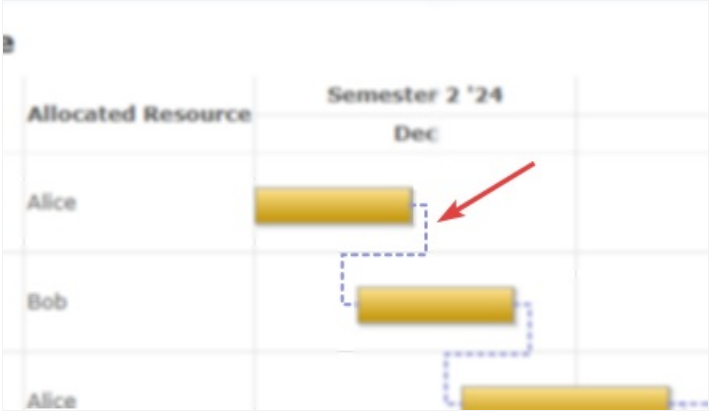
This attribute works together with the **Task ID** field.

Task ID

Defines the field that stores the unique identifier of each activity in the Gantt chart.

It is used in conjunction with the **Predecessor Task** field to create the link between activities and render the visual dependency line.

Example of the line generated between activities:



Milestone Date

Defines a milestone in the schedule, represented as a fixed point in the Gantt chart.

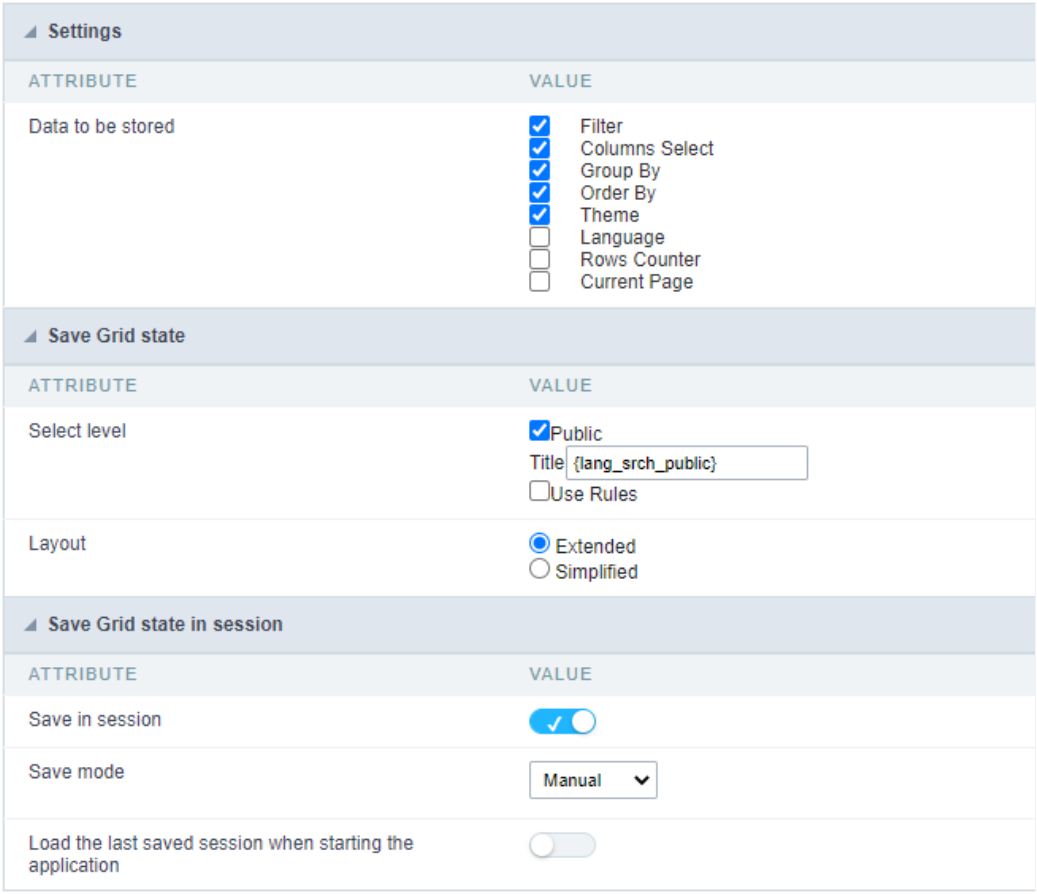
This field must contain the date of an important event or key delivery within the project timeline.

Example of a milestone represented in the chart:

Save Grid

This option allows the user to configure a layout for the consultation and summary application, and save this configuration for later use.

For example, the customer can change the layout and display of fields and change the order of the application, saving this layout. Then, when performing a filter, he can apply the saved layout so that the application is displayed in the way he configured it later.

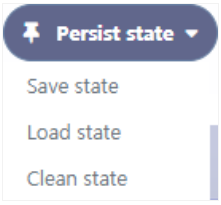


The definition of the data that will be stored by the option Save Grid is defined by the developer.

The Save Grid option can be used in two ways, **Save Grid** and **Persist Status**, the difference between them is the way they are saved.

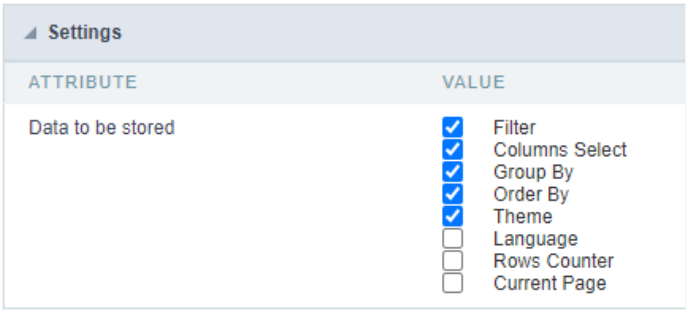
Save Grid - Saves the current grid state to a profile created at the time of saving. Allowing more than one layout to be saved.

Persist Status - Saves the grid state in the session. It is not necessary to create a profile and only stores the last status.



To activate the options, we must add the buttons on the toolbar.

Configuration



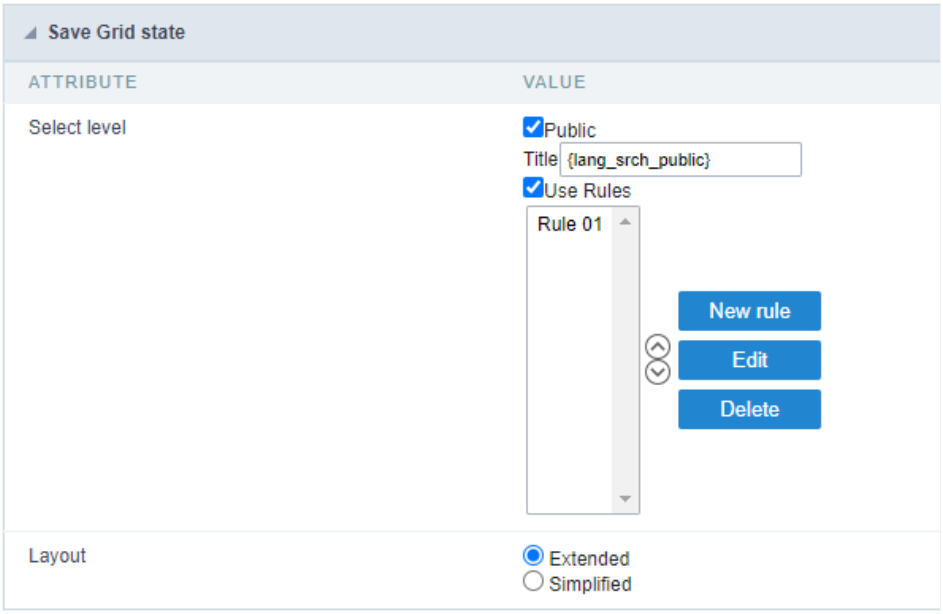
This configuration is related to the two options, **Save Grid** and **Persist Status**, defining the data that will be stored when the option is used.

Below is the list of data that can be stored.

- **Filter**
- **Select Columns**
- **Group By**
- **Order by**
- **Theme**
- **Language**
- **Row Counter**
- **Current page**

The filter option covers all filters in the grid application: Quicksearch, Advanced Filter, Dynamic Filter or Refined Filter.

Save Grid State



These settings refer only to the **Save Grid** option

Set Level

Public - Profiles saved at this level will be visible to everyone who accesses the application.

Use Rules - Creation of rules for profile creation with access restriction. [Click here][link_regra_de_uso_salvar_grid] {:target='blank'} to see how to create a rule.

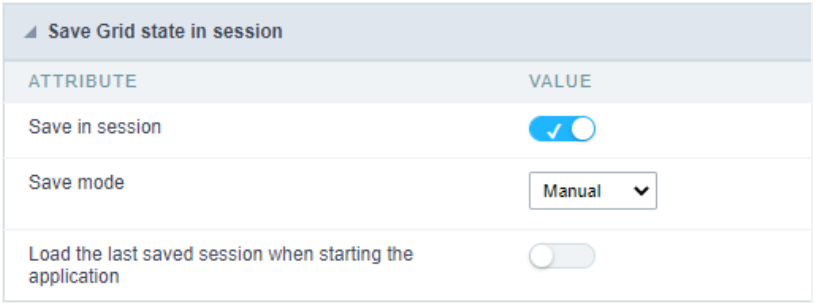
Layout

Defines how to display the **Save Grid** option

Extended - Displays the Save Grid options in a modal below the button.

Simplified - Displays the Save Grid options in a div below the toolbar.

Save Grid state in session



Save in session

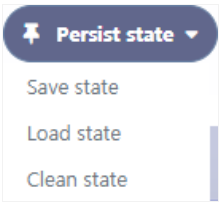
Activates the Persist state option, allowing the saving of the query state in the session, without the need to create a profile.

It will be necessary to add the button **Persistent State** in the application toolbar.

Save mode

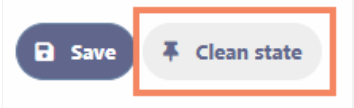
Defines whether saving the state in the session will be carried out automatically or manually. The layout of the **Persistent state** option changes according to the selected option.

Manual - When selected, the option **Persistent State** displays 3 options on the interface.



- Save State - Saves the current grid state in the session
- Load State - Load the last grid state saved in the session
- Clear - Clear the grid state saved in the session

Automatic - When selected, the option **Persistent state** displays only the button **Clear**, as the options **Load State** and **Save State** will no longer be displayed.



Load the last saved session when starting the application

When active, this option causes the application to start with the last state saved in the session. If there is no stored data, the query will start with the configuration defined in its development.

This option is only available when selecting **manual** in the option **Save way**

Related Link 🔗

Related Video ▷

Grid - Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid		
Procedure		
Chart		
Form		
Control		
Calendar		
Search		
PDF Report		

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven't manage fields.

Related Links

Related Video ▷

- QR Code
- Percentage
- Image
- Accumulated
- Pull Down
- Macro sc_text_style
- Macro sc_field_style
- Macro sc_field_color
- Macro sc_field_init_off
- Macro sc_date_empty
- Linking Applications

Grid - Text Field

General Settings

This type of field allows the developer to create quickly fields to display data from the database, where the final user can see the data in the way it was set by the developer.

Type of Data

Defines the type of the field in the application. In this case we should select **Text**.

Label

Defines the title that will be displayed in the field when the application is executed. The terminology used in the interface is fundamental to the system have a good usability, whe should use names and familiar terms to the final user of the application, instead of using terms from the system.

For example, this text field that has the name **customerid**, the client would have a much better understanding of the functionality of the field when we define the label as **Customer Name**.

Not only a fixed text, the **Label** attribute allows the use of langs to define the field's title, making it possible to use your application in a multi language project.

Case Settings

This option allows the developer to inform how the text will be displayed in runtime.

The options available are:

Lower Case: Every letter in the text will be converted to lower case.

Upper Case: Every letter of the text will be converted to upper case.

Capitalize first word: The first letter of the first world will be converted to upper case.

Capitalize all words: The first letter of every words will be converted to upper case.

Show HTML Content

When this option is active every HTML, CSS and JavaScript content that are in the database will be displayed with the main value.

Field Mask

Defines the field mask. There are two types of mask described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

For example, it is possible to set the mast to display a telephone number:

It will be show with this format on runtime:

It is also possible to set the field mask like those examples:

Field mask examples:

Telephone number

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Repeat Value:

When this option is active the value of the field will be repeated if the previous database register is the same.

Example:

SQL Type

Informs the type of the field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).

- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.

- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the

application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Related Video ▷

- Macro sc_text_style
- Macro sc_field_style
- Macro sc_field_color
- Macro sc_field_init_off
- Linking Applications - Field Link

Grid - Multiple Lines Text Field

General Settings

Multiple Lines Text field Configuration Interface.

- **Data Type** : Define the type of field for the application. When it is defined as a Multiple Lines Text, it accepts letters, numbers and special characters in multiple lines.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Case Settings** : Convert the letter from the field when losing focus. The options are:
  - **Upper case** : All in Upper Case
  - **Lower case** : All in Lower case
  - **Capitalize first word** : Capitalizes the first letter of the first word
  - **Capitalize all words** : Capitalizes the first letter of all the the words
- **Show HTML content** : Determines if the HTML contained in the field will be displayed or not. If enabled, the HTML will be displayed, otherwise the HTML will be interpreted by the browser.
- **Grid Mask** : Defines the mask for the field display.There are two typed of masks described below:

Character	Description
X	Placeholder to any character. Replaced by any character. If number of characters entered are less then the mask size, the field value is completed with zeros (Filling full size field entry is required).
Z	Replaced by any character retrieved from database. Suppress zeros at field left (Complete field filling is optional). When used combined with the mask character X it should be placed at the mask left.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

- **Do Not Repeat Value** : Do not repeat the value of the field in case it is the same as the previous record.

- **Run content in JavaScript** : If enabled, the JavaScript will be interpreted by the browser, otherwise the JavaScript will be displayed.
- **SQL Type** : Informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or

bottom).
- **Background Image** : Allows you to add a background image to the field.
- **Line Break** : Enables the possibility of a line break for the field.
- **Content color** : Changes the content color of the query field and the grouped field, when available.
- **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Related Video ▷

- Macro sc_text_style
- Macro sc_field_style
- Macro sc_field_color
- Macro sc_field_init_off
- Linking Applications - Field Link

Grid - Integer Field

On this page, you will learn how you can configure settings related to the Number field. From the use of specific symbols display to the mode in which they are displayed. And thus, boost the application.

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Label:

It sets the title that will be displayed in the field when you run the application. The terminology used in the interface is fundamental for your system has a good usability. In this case, we recommend you to use names and terms familiar to the end user of the application, instead of using terms originated in the system.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Repeat value:

This option when enabled will allows you to repeat the field value if it is equal to the value of the previous record in the database.

Example:

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Color for the negative values:

It allows you to define a color when the value is negative, improving the understanding of the end user about that kind of value.

Example:

c

Display the value in words:

The value of the field will be displayed in full on application. This feature can facilitate the comprehension and understanding of the user.

Example:

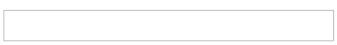
Line size:

Maximum size in characters to be displayed in the value cell, in full. When this value is exceeded the line will break within the cell.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:



Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set

in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:
- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

  In the example shown above, the width of the Chart is 200 pixels.
- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.

- **Display Value** : It will display the value and the chart at the same time, the value being on the left or right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Charts Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Related Video ▷

- Macro sc_trunc_num
- Macro sc_format_num_region
- Macro sc_format_num
- Macro sc_text_style
- Macro sc_field_style
- Macro sc_field_color
- Macro sc_field_init_off

Grid - Decimal Field

General Settings

Decimal field Configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the

select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.

- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:

- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

  In the example shown above, the width of the Chart is 200 pixels.

- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.

- **Display Value** : It will display the value and the chart at the same time, the value being on the left or right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this

field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Related Video ▷

- Macro sc_trunc_num
- Macro sc_format_num_region
- Macro sc_format_num
- Macro sc_text_style
- Macro sc_field_style
- Macro sc_field_color
- Macro sc_field_init_off

Grid - Percent Field

General Settings

Percentage field Configuration Interface.

Data Type

Define the field type to Percentage.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value

returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and

lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:
- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

  In the example shown above, the width of the Chart is 200 pixels.

- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.

- **Display Value** : It will display the value and the chart at the same time, the value being on the left or right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not

informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Related Links

- Percentage

Related Video

- Macro sc_text_style
- Macro sc_field_style
- Macro sc_field_color
- Macro sc_field_init_off

Grid - Currency Field

General Settings

Currency field Configuration Interface.

Data Type

Define the field type to Currency. It allows defining the currency number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	Placeholder to any character. Replaced by any character. If number of characters entered are less then the mask size, the field value is completed with zeros (Filling full size field entry is required).
Z	Replaced by any character retrieved from database. Suppress zeros at field left (Complete field filling is optional). When used combined with the mask character X it should be placed at the mask left.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

![Displaying repeated values.][cons_configuracao_geral_texto_nao_repetir_valor]

SQL Type

It informs the data type of field in the database.

Values Format

Interface of Values Format.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Currency Format

It defines if the field displays the Currency Symbol of the Regional Settings.

Currency Symbol

It allows setting the character that represents the Currency Symbol.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Display full value

It displays the full value. Example: 2018(Two thousand eighteen).

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the "Separated by" field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.

- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:

- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

In the example shown above, the width of the Chart is 200 pixels.

- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.

- **Display Value** : It will display the value and the chart at the same time, the value being on the left or right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Related Video ▷

- Macro sc_trunc_num
- Macro sc_format_num_region
- Macro sc_format_num

- Macro sc_text_style
- Macro sc_field_style
- Macro sc_field_color
- Macro sc_field_init_off

Grid - Percent (Calculated) Field

General Settings

Percentage field Configuration Interface.

Data Type

Define the field type to Percentage.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value

returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and

lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:
- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

  In the example shown above, the width of the Chart is 200 pixels.

- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.

- **Display Value** : It will display the value and the chart at the same time, the value being on the left or right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not

informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Related Links

- Percentage

Related Video

- Macro sc_text_style
- Macro sc_field_style
- Macro sc_field_color
- Macro sc_field_init_off

Grid - HTML Image Field

General Settings

HTML Image field Configuration Interface.

Data Type

Define the type of field. When setting it to HTML Image, it allows to display an image into the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Image

Define an image to be displayed. The icon “Select Image” lists all images from Scriptcase and your uploaded images. The “Upload an image” option allows you to send a copy to the Scriptcase server.

New Link

This button allows the creation of a field link with some other application..

This way, it is possible, for example, to create a link with a blank to delete a record from the grid, passing the record ID as a parameter.

> See more information about field binding by clicking here.

Border

Define the width of the Image border in Pixels.

Width

Define the image width size in Pixels.

Height

Define the image height size in Pixels.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the

background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Related Links 🔗

- Image

Related Video ▷

- Linking Applications - Link Field
- Macro sc_field_style
- Macro sc_field_init_off

Grid - Credit Card Number Field

General Settings

Credit Card Number Configuration Interface.

- **Data Type** : Select the type of field for the application. When it is defined as a Credit Card Number, you can define some rules for the display format of the Credit Card.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **Do Not Repeat Value** : Do not repeat the value of the field in the case it is the same as the previous record.

- **SQL Type** : Informs the data type of the field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Related Video ▷

- Macro sc_text_style
- Macro sc_field_style
- Macro sc_field_color
- Macro sc_field_init_off
- Linking Applications - Field Link

Grid - E-mail Field

General Settings

Email field Configuration Interface.

- **Data Type** : Defines the type of field for the application. When it is defined as an Email, when you click on the field you be offered a choice for your email client and send an email to that specific email.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Do Not Repeat Value** : Do not repeat the value of the field in case it is the same as the previous record.

- **SQL Type** : Informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Grid - URL Field

General Settings

URL field Configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Grid - YouTube Field

General Settings

YouTube field Configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Display Mode

Define the mode to display the video in the application.

Width

Video Width in pixels.

Height

Video Height in pixels.

Link Type

The display settings of the link, if it is going to be a Text or a Button.

Link text

It displays a text to mask the video URL.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.

- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Grid - Google Maps Field

General Settings

Google Maps field Configuration Interface.

- **Data Type** : Define the type of field for the application. When it is defined as Google Maps, it will use the Google Maps API to display the map in the Grid Applications
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Home** : Defines what type of parameters will be used in the API.

Home Configuration Interface.

- **Display Mode** : Indicates the display mode of the map. It can be opened in a Modal or in a new Window.
- **Width** : Defines the width of the map that is going to be displayed.
- **Height** : Defines the height of the map that is going to be displayed.
- **Zoom** : Defines the initial Zoom (available from the Google API) of the Map location.
- **API Key** : API Key for authorization to use Google Maps in the Application. (Required only for the versions 2 or earlier of the Google API.)

  The API Key is a unique key, composed by a string(text) alphanumeric, which is the license to use the service. When you subscribe to use the service, the key is tied to the domain and the directory of the server. All the pages that use the API needs to be in the same directory that was used for the subscription. In case you have a web server on your local machine, you just need to possess a key for testing, and to do this you only need to set http://localhost in the domain of the subscription.

  To get your API Key access the site by clicking here

- **Link Type** : Defines how the link will be displayed.
- **Text Link** : Text to call the Map.

- **Marker Description** : Displays the description for each marker displayed on the map.

- **SQL Type** : Informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top,

middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Grid - Date Field

General Settings

Date field Configuration Interface.

Data Type

Define the type of field. When setting it to Date, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

 Format of Values without Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying dates.

Detail Mask

Define the format of the field in the Grid. It uses the patterns of PHP function Date.

d-m-Y	25-09-2009
F/Y	September/2009
j/n/Y \a\s g:i:s A	25/9/2001 as 14:30:11 PM
l, d \d\e F \d\e Y	Thursday, 25 of January of 2009
h:i:s	11:33:20
#h:i:s	123:43:27 (accumulating the hours)

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

- **Example 1** : If the field is storing the Date value in a **char(8)** where the positions **1** to **4** represents the year, the positions **5** and **6** is the month and the position **7** and **8** is the day, you must define the internal format as: **YYYYMMDD**
- **Example 2** : If the field is storing the Date value in a **char(6)** where the positions **1** and **2** represents the month and the positions **3** to **6** is the year, you must define the internal format as: **MMYYYY**

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Grid - Time Field

General Settings

Time field Configuration Interface.

Data Type

Define the type of field. When setting it to Time, you can inform a time format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

 Format of Values without Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the Time. When not enabled, it displays the Time separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying Time.

Detail Mask

Define the format of the field in the Grid. It uses the patterns of PHP function Date.

h:i:s	11:33:20
#h:i:s	123:43:27 (accumulating the hours)

Internal Format

It allows defining the field format when the SQL type is different from **TIME**. You must use the characters **HH**, **II**, and **SS** that correspond to **Day**, **Hour**, **Minutes** and **Seconds**.

- **Example 1**: If the field is storing the Date value in a **char(8)** where the positions **1** to **4** represents the year, the positions **5** and **6** is the month and the position **7** and **8** is the day, you must define the internal format as **YYYYMMDD**
- **Example 2**: If the field is storing the Date value in a **char(6)** where the positions **1** and **2** represents the month and the positions **3** to **6** is the year, you must define the internal format as **MMYYYY**

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of

scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Grid - Date and time Field

General Settings

Date and Time field Configuration Interface.

Data Type

Define the type of field. When setting it to Date and Time, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

 Format of Values without Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying dates.

Detail Mask

Define the format of the field in the Grid. It uses the patterns of PHP function Date.

d-m-Y	25-09-2009
F/Y	September/2009
j/n/Y \a\s g:i:s A	25/9/2001 as 14:30:11 PM
l, d \d\e F \d\e Y	Thursday, 25 of January of 2009
h:i:s	11:33:20
#h:i:s	123:43:27 (accumulating the hours)

![Date Format Table][cons_data_format_date]

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

- **Example 1** : If the field is storing the Date value in a **char(8)** where the positions **1** to **4** represents the year, the positions **5** and **6** is the month and the position **7** and **8** is the day, you must define the internal format as: **YYYYMMDD**
- **Example 2** : If the field is storing the Date value in a **char(6)** where the positions **1** and **2** represents the month and the positions **3** to **6** is the year, you must define the internal format as: **MMYYYY**

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

PDF Configuration

Line Break

This option allows breaking the line in the PDF file when exported.

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Grid - Image (Database) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Image (Database), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Image Border

Width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Open in Another Window

Allows to open the image in another window.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.

- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Grid - Image (File Name) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Image (File Name), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Image Border

Width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API.

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API.

Open in Another Window

Allows to open the image in another window.

Subdirectory for local storage

Local subdirectory or on the server where the files are stored. eg: {CustomerId} , [glo_var_seq]

Image Caching

Time in minutes the image cache will be kept in the server before being deleted.

Repeat value

Repeat the field value if it is equal to the previous record.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Grid - Document (Database) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Document (Database), all the document files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Icon

Displays an icon beside the field to identify the type of document.

File Name

It allows defining the field to store the name of the document in the database.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Grid - Document (File Name) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Document (File Name), all the document files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Sub-folder

Sub-folder name that the files are stored.

Icon

Displays an icon beside the field to identify the type of document.

File Name

It allows defining the field to store the name of the document in the database.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API.

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or

bottom).
- **Background Image** : Allows you to add a background image to the field.
- **Line Break** : Enables the possibility of a line break for the field.
- **Content color** : Changes the content color of the query field and the grouped field, when available.
- **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Grid - Bar code Field

General Settings

Configuration Interface of the Barcode Field.

- **Data Type** : DataType of the field for the application.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **SQL Type** : Database field type.

Watch below a video showing an example about the Barcode field:

Values Format

Configuration Interface of the Barcode Field.

- **Type** : Type of Barcode.
- **Text** : Barcode Text for illustration purposes.
- There are **18 types of barcodes**, that are listed below:

Barcode configuration interface.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.

- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Related Links 🔗

- Barcode

Related Video ▷

- How to create and use barcodes
- Macro sc_field_style
- Macro sc_field_init_off

Grid - QRCode Field

General Settings

QRCODE field Configuration Interface.

Data Type

Define the type of field. When setting it to QRCODE, it allows you to transform values into a QRCODE.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

SQL Type

It informs the data type of field in the database.

Watch below a video showing an example with QRCODE

Values Format

Interface of Values Format.

Level of error correction

The Codewords are 8 bits long and use the Reed–Solomon error correction algorithm with four error correction levels. The higher the error correction level, the less storage capacity.

Image Size

Set the size of the QRCODE image.

Margin

Set the margins of the QRCODE.

 Interface of Values Format.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.

- **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
- **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
- **Background Image** : Allows you to add a background image to the field.
- **Line Break** : Enables the possibility of a line break for the field.
- **Content color** : Changes the content color of the query field and the grouped field, when available.
- **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Related Links 🔗

- QR Code

Related Video ▷

- Working with QR codes
- Macro sc_field_style
- Macro sc_field_init_off

Grid - Signature Field

General Settings

Configuration Interface of the Signature Field.

The signature field will help you creating more sofisticated forms and making it possible to store signatures in your database. Inside our development environment we have specific settings that will help you to customize your field, those options are:

- **Data Type** : You can define the type of field for the application. When it is defined as a text, it accepts letters, numbers and special characters.
- **Label** : Lets you define a label to the field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Background color** : Defines a color to the field background by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Pen color**: Set a color to the pen by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Width** : Lets you define a width to the field.
- **Height** : Set a height to the field.
- **Subtitle** : Defines the subtitle that will be displayed beside the field.
- **Initial Value** : Lets you define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
- **Save Variable** : Save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Disabled Field** : Define if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : Displays the HTML Object that will be used to display the field in the form.
- **SQL Type** : Displays the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of ScriptCase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **CSS of the Title**
  - **Font** : Select the font type, that will be applied to the application field title.
  - **Font Size** : Defines the font size, that will be applied to the application field title.
  - **Font Color** : Choose a color to the font by using a valid hexadecimal color value or from the color picker.
  - **Background Color** : You can define the color for the field by using a valid hexadecimal color value or from the color picker.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Set the underline style to the font.
  - **Border style** : Choose a style for the border.
  - **Collapse** : Defines the collapse for the border.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Choose a color for the border, using a color palette to apply to the title.
  - **Horizontal Alignment** : Position the label of the field in the wanted position (left,right,center and justify).
  - **Vertical Alignment** : Position the label of the field in the wanted position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : Defines a width for the title of the field.
  - **Height** : Set a height for the title of the field.

Grid - Rating Field

This field allows the developer to create a field to display the data previously added using stars (or any other image).

Attribute descriptions

Data Type

Define the field type to **Rating**.

Label

The Label option lets you define the title of a field. Example: If the database field name is “**Stars**”, You can display a different name for the user, like “**Stars**”.

Besides use a fixed text, the **Label** attribute allows the use of **Langs** to define the field title, allowing the internationalization of your application.

Subtitle

Define the subtitle of the field, below the ratings. **Example**: “Thank you for your feedback!”.

As in the **Label**, the **subtitle** attribute also allows the use of **Langs** for the internationalization of your application.

Amount of icons

It defines how many icons it will display in the field. The value set in this attribute must be according to the evaluation rules.

Use in Group by and Sums

Allows the field to be used instead of the value when the field is selected in the **(static or dynamic)** group by or in the totalization of the summary. To do this, just add the fields in the dynamic or static group by column to be able to visualize the grouping performed in the query.

Example of the **Rating** field in the summary.

In the summary, when hovering the mouse over the field, a small window detailing the ratings that were carried out both in each record and in total will be displayed, as shown in the image below.

Display absolute value

Allows the exact number of ratings that each star received to be displayed in the detailing window, next to the percentage, when summing up the summary.

> With this flag turned off, the grouping will only display the voting percentage, leaving out the exact number of ratings.

In the case of the result of the ratings, this result is calculated through the weighted average **(Sum/Total of Evaluations)** as in the example below:

For example: in the image above, we have a total of 4 ratings carried out, where:

0 rating for 5 stars
2 ratings for 4 stars

0 rating for 3 stars
1 rating for 2 stars
1 rating for 1 star

Considering that the number of stars refers to the weight each one has, we have the following:

5 stars = 0 rating x 5 (weight) = 0
4 stars = 2 ratings x 4 (weight) = 8
3 stars = 0 rating x 3 (weight) = 0
2 stars = 1 rating x 2 (weight) = 2
1 star = 1 rating x 1 (weight) = 1

Sum = 11

We took the result of the sum and divided it by the total number of evaluations, which was 4.

Sum/Total of Evaluations = 11/4 = 2.75 ≈ 2.8.

As only 1 value after the comma will always be displayed in the detail window, rounding will occur upwards when the values after the comma are above the central value, and downwards when it is less than the central value.

Ex.: 2.75 ≈ 2.8.

Use Font Awesome

Allows you to use Font Awesome icons to represent stars instead of images.

Rating Star Hints

Defines the hint for each star

Icon size

Sets the pixel size of star icons.

Rating Padding

Defines the padding value in pixels for separating each star.

Icons Color

Allows you to define the color of the stars; the color is passed in hexadecimal.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position

(left,right,center and justify).
    - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the title of the field.
    - **Height** : To define a height for the title of the field.
- **CSS of the Field**
    - **Font** : Allows to choose the font type, that will be applied to the application field.
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the field.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
    - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the field.
    - **Height** : To define a height for the field.
- **CSS of the Input Object**
    - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the Input Object.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
    - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the Input Object.
    - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Grid - Rating Smile Field

General settings

This field type allows the developer to show data as an evaluation with emoji that goes from **Very Bad** to **Excellent**, where the final user could see the last evaluation.

Data type

Defines the data type of the field in the application. In this case we should select **Rating Smile**.

Label

Defines the title that will be shown in the field when the application is running. The used terms in the interface are essential to the system has good usability, we should use comum name and terms for the final user, instead of the default values of the system.

For example, for the rating smile field, the client would have a better understanding of the field function when the label is **Product evaluation**.

Besides a fixed text, the **Label** allow to use of lang to define the field title, this way is possible to use the application internationalization.

Subtitle

Define a subtitle that will be shown beside the field. Ex.: Thank you for the feedback!

Below is one example of an application subtitle:

Use in Group by and Sums

Allows the field to be used instead of the value when the field is selected in the **(static or dynamic)** group by or in the totalization of the summary. To do this, just add the fields in the dynamic or static group by column to be able to visualize the grouping performed in the query.

Example of the **Rating Smile** field in the summary.

In the summary, when hovering the mouse over the field, a small window detailing the evaluations that were carried out both in each record and in total will be displayed, as shown in the image below.

Display absolute value

Allows the exact number of evaluations that each icon received to be displayed in the detailing window, next to the percentage, when summing up the summary.

> With this flag turned off, the grouping will only display the voting percentage, leaving out the exact number of ratings.

In the case of the result of the evaluations, this result is calculated through the weighted average **(Sum/Total of Evaluations)** as in the example below:

Excellent: 5 ratings - 16.7%
Good: 7 ratings - 23.3%
Regular: 11 ratings - 36.7%
Bad: 3 ratings - 10%

Very bad: 4 reviews - 13.3%

Considering that the weight of each smile always varies from 1 to 5 (Very bad to Excellent), we have:

5 ratings x 5 (weight) + 7 ratings x 4 (weight) + 11 ratings x 3 (weight) + 3 rating x 2 (weight) + 4 ratings x 1 (weight)

Sum = 96 —> Result = 96 / 30 (total evaluations) = 3.2

As the result taken into account in the **Rating Smile** field is always an integer, and the value after the decimal point is below the central value (in this example, the central value would be 3.2), so we rounded DOWN, and we got the same result a 3, which corresponds precisely to the **“Regular”** smile, and that is why this is the one that is most lit at the top of the detailing window, as shown in the image above.

Values and hints

Allows to the developer to define the values to each emoji taht will be saved at the data base and allow to define a hint **(It is show passing the mouse above the emoji)**

Display format

Allows the developer to define the format that the emojis will be shown.

In case the first option is defined, the emojis will be shown like this:

In case the second option is defined, the emojis will be shown like this:

Icon size

Allows defining the icon size in pixels.

Rating padding

Allows for defining the padding value between the emojis.

Icon colors

Allows the icon colors to be defined in each emoji.

SQL type

Infomrs the field type at the data base.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).

- **Background Image** : Allows you to add a background image to the field.
- **Line Break** : Enables the possibility of a line break for the field.
- **Content color** : Changes the content color of the query field and the grouped field, when available.
- **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Grid - Rating Thumbs Field

General settings

This field type allows the developer to show datadata as an evaluation with icons “like” and “dislike”.

Data type

Defines the data type of the field in the application. In this case we should select **Rating Thumbs**.

Label

Defines the title that will be shown in the field when the application is running. The used terms in the interface are essential to the system has good usability, we should use comum name and terms for the final user, instead of the default values of the system.

For example, for the rating thumbs field, the client would have a better understanding of the field function when the label is **Product evaluation**.

Besides a fixed text, the **Label** allow to use of lang to define the field title, this way is possible to use the application internationalization.

Subtitle

Define a subtitle that will be shown beside the field. Ex.: Thank you for the feedback!

Below is one example of an application subtitle:

Use in Group by and Sums

Allows the field to be used instead of the value when the field is selected in the **(static or dynamic)** group by or in the totalization of the summary. To do this, just add the fields in the dynamic or static group by column to be able to visualize the grouping performed in the query.

Example of the **Rating Thumbs** field in the summary.

In the summary, when hovering the mouse over the field, a small window detailing the evaluations that were carried out both in each record and in total will be displayed, as shown in the image below.

Display absolute value

Allows the exact number of evaluations that each thumb received to be displayed in the detailing window, next to the percentage, when summing up the summary.

> With this flag turned off, the grouping will only display the voting percentage, leaving out the exact number of ratings.

As there are only two possible options to choose from, only records are counted to define the percentage of times each thumb was selected.

For the scenario in which both thumbs have the same number of ratings, with a percentage of 50% x 50%, the thumb that will be more lit at the top of the detailing window will always be the “Like”.

Values and hints

Allows to the developer to define the values to each icon taht will be saved at the data base and allow to define a hint **(It is shown passing the mouse above the icon)**

Display format

Allows the developer to define the format that the emojis will be shown.

In case the first option is defined, the emojis will be shown like this:

In case the second option is defined, the emojis will be shown like this:

Icon size

Allows defining the icon size in pixels.

Rating padding

Allows for defining the padding value between the emojis.

Icon colors

Allows the icon colors to be defined in each emoji.

SQL type

Informs the field type at the data base.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.

- **Bold** : Displays the field title in bold

Grid - Fields Grouping

This feature allows to group two or more fields into a virtual field of the special type **Field Grouping**.

> See how you can create a Field Grouping.

Fields Grouping example

General Settings

See below the option's general configuration for fields grouping.

Data Type

Informs the data type of the field in the application.

Label

Defines the *title* displayed in the application column.

The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.

> See more details about langs and about our Data dictionary.

Grouping fields

In this attribute we define the fields that will be grouped, moving them to the right, and the order in which these fields will be displayed on the grouping fields.

Display options

Defines how grouped field values will be display in the field.

When selecting the option *Label and Value* (Default value) defines that both the **label** and the **field value** will be displayed, see the example below.

When selecting the *Value* option, only the field value will be displayed, see the example below.

Label positioning

- **Above** - Default value, positions the label of the grouped fields above the values.
- **Beside** - Positions the label of the grouped fields beside the values.

  This attribute is only available when selecting the *field value* option in the **Display Options** attribute.

Example of above label

Example of beside label

Spacing beteween fields

Sets the spacing between grouped fields. This attribute is applied in pixels, for this definition we must inform only the numerical value.

Examples of fields with and without spacing

Title for sorting grouped fields

This attribute defines the title of the grouping sort box.

> By default, Scriptcase defines the lang {lang_field_grouping_choose_column_sorting} as the title. The lang text can be changed in the tool Application Language.

Example of sorting grouped fields

SQL type

Defines the SQL type of the field.

Display Settings

See below the options to display settings of the field grouping.

Title horizontal Alignment

In this attribute you can set the horizontal alignment of the field grouping title as: **Left**, **Right** or **Centered**

> When leaving this value unchanged, the alignment assumes the configuration of the application's theme.

Examples of horizontal alignments: Left - Right - Centered

Title vertical alignment

In this attribute you can set the vertical alignment of the field grouping title as: **Top**, **Middle** or **Bottom**

Examples of vertical alignments: Top - Middle - Bottom

Backgroud color

Defines the background color of the entire column of the grouping of fields.

> The configuration of grouped fields must be carried out individually in the field.

Title color

Set a color to field grouping title overriding theme settings.

Example of label text color

Field title backgroud color

Set a background color to field grouping title overriding theme settings.

Example of the background color of the label

Bold

Defines whether or not the button group title will be in bold.

Grid Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Blocks

Blocks are “containers” where you can position the application fieldSlides of Forms, Controls, or Grids.

Scriptcase creates applications with one block by default. You can add more blocks as you wish, to organize it in the best way.

See below, the Columns Organization, and where you can define the position of the next block: beside or below the current one.

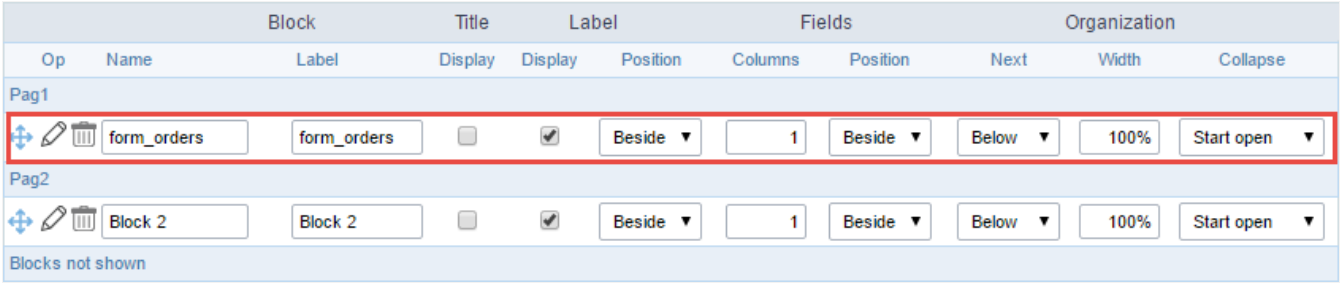
Application Block configuration

On the left side of each block, there are two icons, the first one to edit the information of the block and the second one to delete the block.

Organizing the position of the Blocks

See below how to modify the display order of the Blocks in one Page.

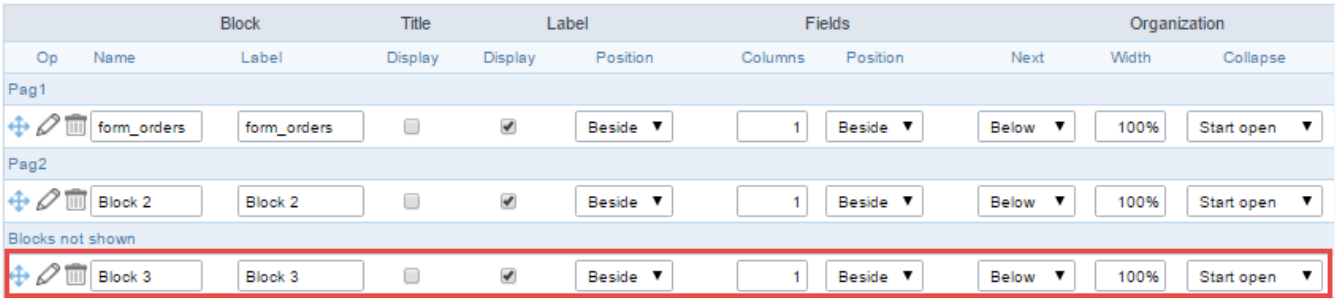
Click and drag the block that you desire to modify to its new position.



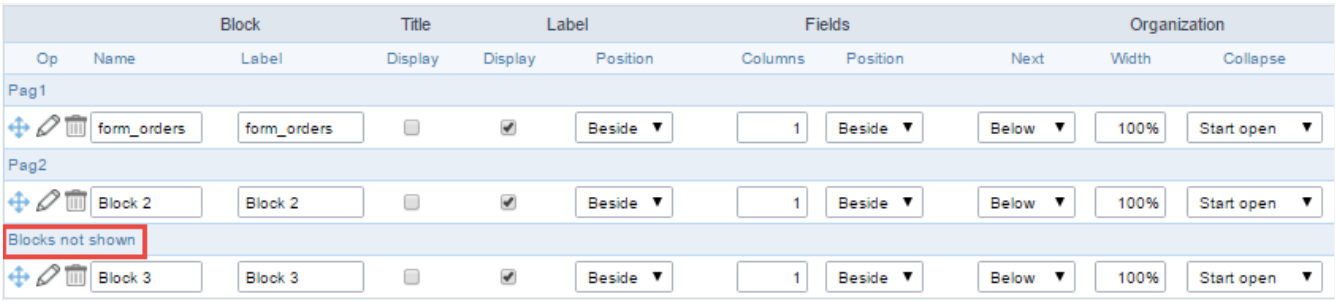
Application Block Display configuration

- See how to remove a block from the display

Click on the block desired and drag it to the item “Blocks not Shown”. This way, you can also drag the block to another page if desired. See the images below.



Application Block Display configuration



Application Block Display configuration

Block

- **Name**: The name that identifier the Block.
- **Label**: Title of the block to display in the application.

Title

- **Display**: It controls the display of the block title.

Label

- **Display**: It controls the display of the field labels of the block.
- **Position**: Options to display label :
    1. **Above**: Display the label above the field.
    2. **Beside**: Display the label beside the field.
    3. **Below**: Display the label below the field.

Fields

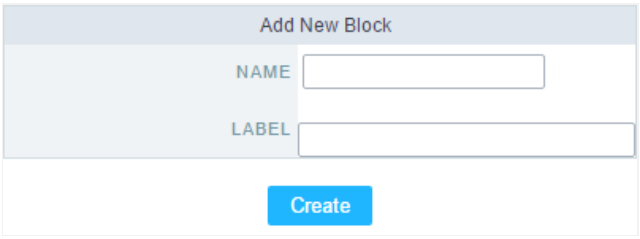
- **Columns**: Amount of columns side by side in the block.
- **Position**: The way to display the fields in the block:
    1. **Below**: Display the fields one below the other respecting the number of columns.
    2. **Beside**: Display the fields one beside the other respecting the number of columns.
    3. **Line**: Display the fields one beside the other with no tabulation.

Organization

- **Next**: The way to display the blocks in the page:
    1. **Below**: Set to show the following block below the current one.
    2. **Beside**: Set to show the following block beside the current one.
    3. **Tabs**: Set to show the following block in a different tab then the current one.
- **Width**: Set the block width in pixels or percentages. Use the symbol “%” to indicates the value in percentage.
- **Collapse**: Enables the option to close the block.

Create a New Block

To include new blocks in an Application, click on the button Create New Block. Then, enter the name and label of the block in the following interface and finish by click on Create.



Creating application blocks configuration

Name

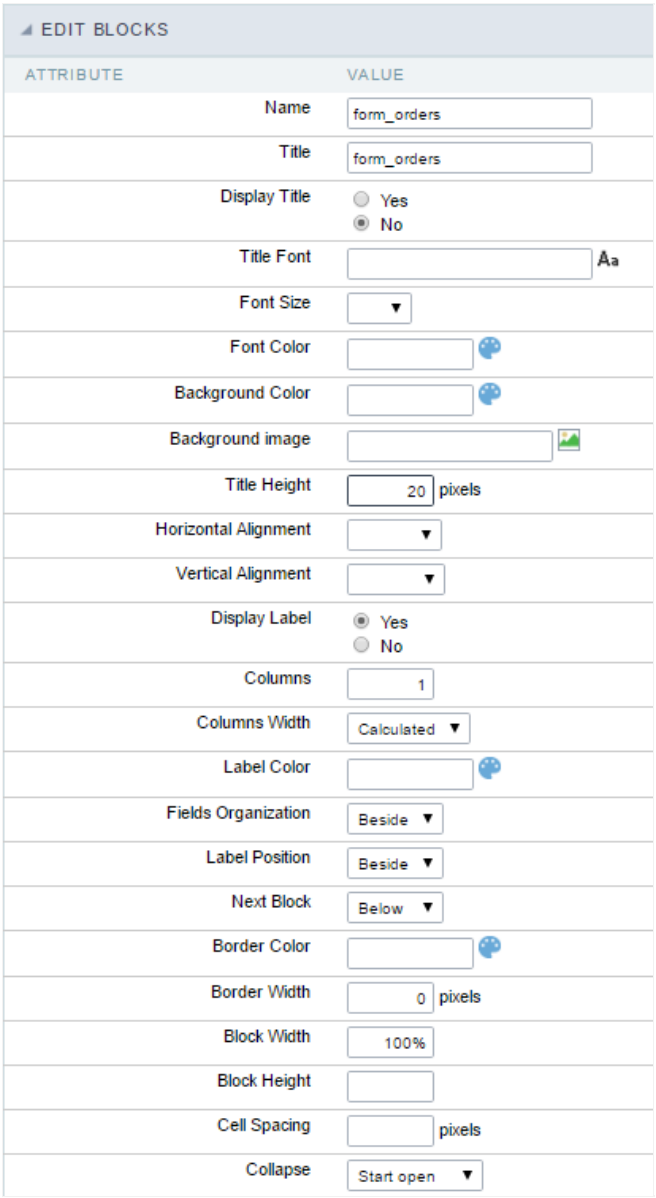
Name of the Block.

Label

Title of the block to display in the application.

Edit Blocks

To edit a block, click on the icon , that is on the left side of the block. Then you can see the following interface to define the parameters of the blocks. Click on Save to finish.



Application Block editing interface

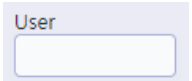
Name

Name of the block. #### Title
Block title for display. #### Display Title
This option, when active, allows displaying the block title. #### Title Font
Set the font family of the block title. #### Font Size
Set the font size of the block title. #### Font Color
Set the font color of the block title. #### Background Color
Set the Background Color of the block title. #### Background image
Set a Background image for the block title. #### Title Height
Height in pixels of the block title line. #### Horizontal Alignment
Horizontal Alignment of the block title (Left, Center, and Right). #### Vertical Alignment
Vertical Alignment of the block title (Top, Middle, and Bottom). #### Display Label
Display the labels of the fields in the block. #### Columns
Amount of field columns in a block. #### Columns Width
Set the field column width of the block. #### Label Color
Color of the field labels. #### Fields Organization
The way to display the fields in the block. #### Label Position
Set the position of the field labels of the block.

The options are:

- **Beside** - This option positions the label on the right side of the field.

- **Above** - This option places the label above the field.



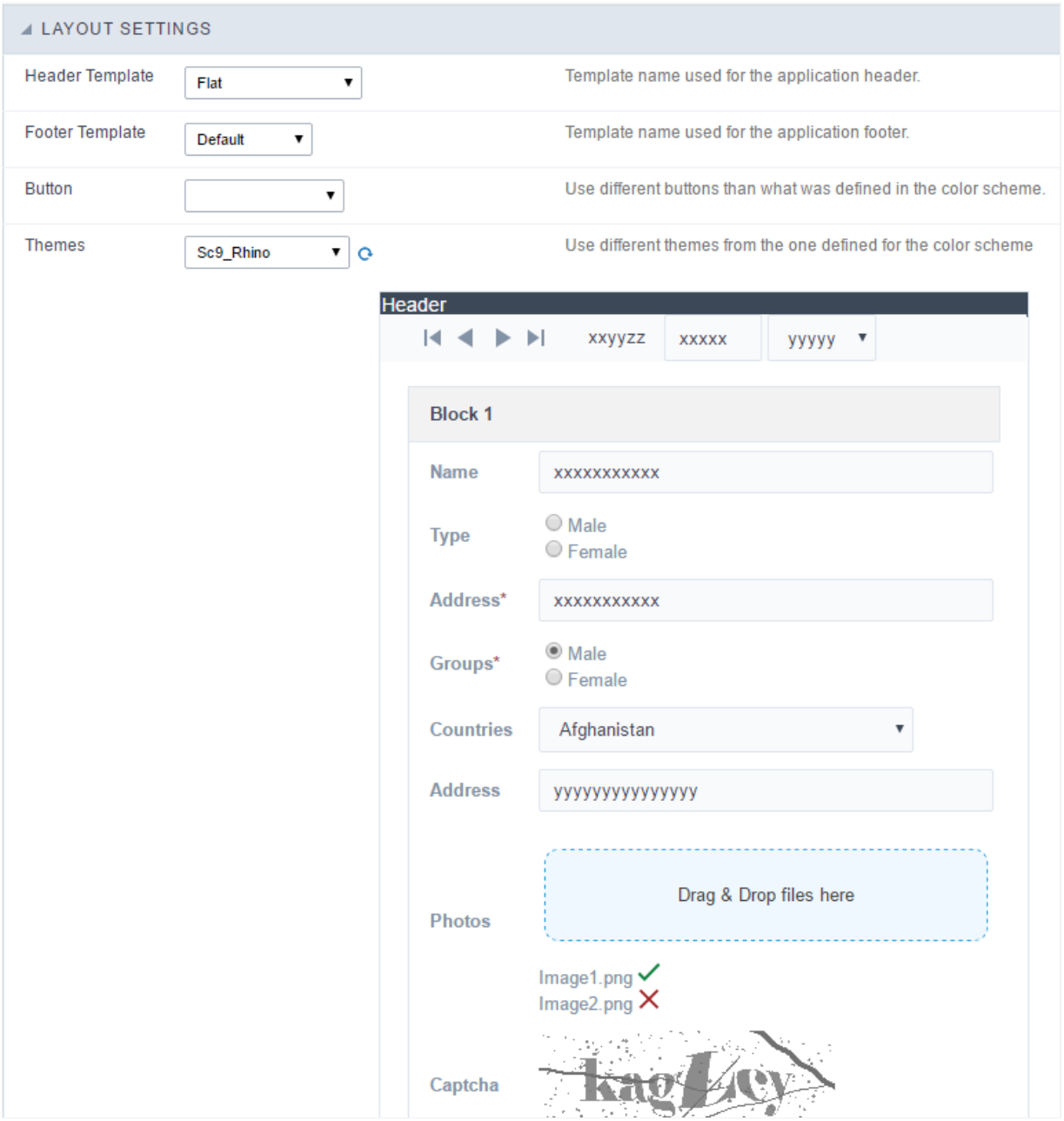
- **Below** - This option places the label below the field.

Next Block

Set the position of the following block relating to the current one. #### Border Color
The border Color for the block. #### Border Width
The border Width for the block. #### Block Width
The width for the block. #### Block Height
The Height for the block. #### Cell Spacing
The Cell Spacing in the block. #### Collapse
It enables the option to close the block.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

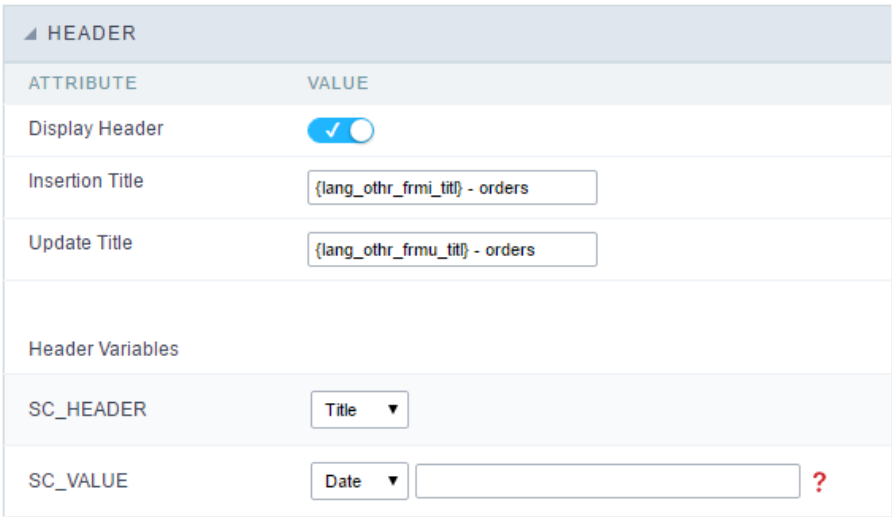
Allows choosing the button theme for the application.

Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

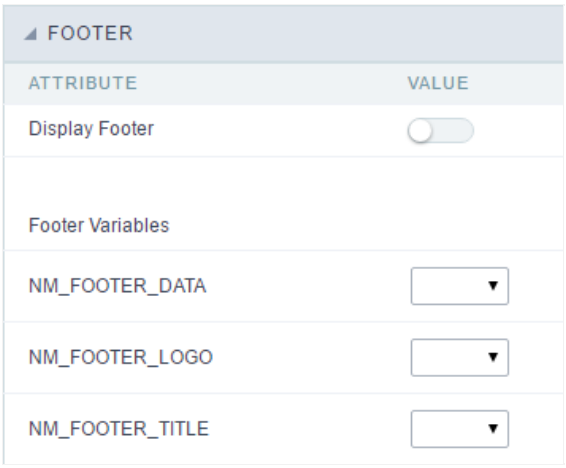
- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the

icon “**Choose Image**”, and you still can upload new images by using the button “**Upload**”. .

- **Value**: It displays the content of the text input. You can inform static texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option “**Field**”, it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system’s date in “yyyy/mm/dd” format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon “**Choose Image**”, and you still can upload new images by using the button “**Upload**”. .
- **Value**: It displays the content of the text input. You can inform static texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

> **Related Links**

Grid Events Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application's CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event's PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Grid

It is the first event executed in the application, being fired even before the SQL and HTML assembly.

It works as a preparation for the application, where it is possible, among other things, to manipulate variables, perform validations and change the connection with the sc_change_connection macro, for example.

In the example below, access to the edit form (form_orders) linked to the query will be limited, depending on the user who is accessing the application.

If the variable [usr_login] is equal to **admin**, the user will have access to the form with the possibility of inserting and deleting a record.

If the variable has some other value, the user will still have access to the form, but only with the possibility of changing a record.

Samples Code

```
if ( [usr_login] == 'admin') {

  sc_apl_conf("form_orders", "insert", "on");
  sc_apl_conf("form_orders", "delete", "on");

} else {

  sc_apl_conf("form_orders", "insert", "off");
  sc_apl_conf("form_orders", "delete", "off");

}
```

onNavigate - Grid

In the query application, this event runs in two situations:

- When configured with partial paging, the event is executed when navigating between pages using the **navigation buttons**, **navigation by page** or the **Jump to** option.
- When configuring the application with **Infinite Scroll**, it is executed every time records are loaded.

Unlike other events, this one depends on user interaction to be triggered, allowing, for example, that any validation or layout change is performed only after this interaction.

> **Note**
>
> The query navigation is performed by ajax, thus, it is not possible to use Javascript codes (JQuery, Ajax).
>
> When using full paging in the application, the event is not executed.

onScriptInit - Grid

Second event to be executed, before the execution of the main select, it is triggered whenever the application is loaded. For example, after using the advanced filter.

In it, all application variables and libraries are available for use.

The manipulation of connections with sc_connection_edit and sc_connection_new macros, manipulation of the main select with sc_select_order and sc_select_where(add) macros and inclusion of libraries already incorporated into Scriptcase such as Jquery with sc_include_lib macro are some of the examples of using the event.

In the example below, access to the sales report will be limited to the user who performed them.

Access is limited if the user is not the admin.

Sample Code

```
if ( [usr_login] != 'admin' ) {

   if ( empty({sc_where_atual})){

      sc_select_where(add) = "WHERE employeeid > [usr_login]";

   } else {

      sc_select_where(add) = "AND employeeid > [usr_login]";
   }
}
```

onRecord - Grid

This event is executed immediately before the display of each record displayed in the query, regardless of user interaction.

Thus, it is commonly used for data manipulation and validation, creating links with the sc_link macro or changing the layout with the sc_field_style macro, based on the displayed information.

In the example below, the style of the text in the {priceorder} field will change according to the value.

For values below 500 reais, the text will have a different color from values above 500 reais,

Samples Code

```
if ( {priceorder} < 500 ) {

  sc_field_style({priceorder}, "Background-Color", "15", "#228B22", "", "bold");

} else {

  sc_field_style({priceorder}, "Background-Color", "15", "#006400", "", "bold");

}
```

onHeader - Grid

The third event generated in the query occurs immediately before the application's header display, during the loading of the page's display HTML elements.

> The onHeader is only obtained when the option to display the header is enabled in the application's layout settings.

In this event, it is possible to perform queries in the database, allowing you to read information about the application and the values of the Query's fields.

It is used, among other things, to change the application's CSS, javascript codes and display information linked to fields, such as a sales total or a legend for a line graph.

onFooter - Grid

This event occurs after the processing of the grid lines, during the display of the HTML elements of the page footer.

onFooter is used when we want to display some information in the footer, such as a sales total or a legend for a line graph.

> onFooter runs only when the option to show footer is enabled in the application's layout settings.

E.g: We want at the bottom of the application to display the total value of orders displayed in the grid of an invoice, with a 10% discount:

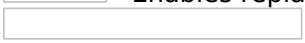
```
sc_lookup(ret," select sum( valueitem ) from sales_item where sale_id={sale_id} ");

[v_footnote] = {ret[0][0]} * 0.1;
```

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

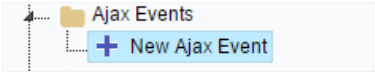
> Click Here to view the Scriptcase hotkeys documentation.

Grid Ajax Events

OnClick

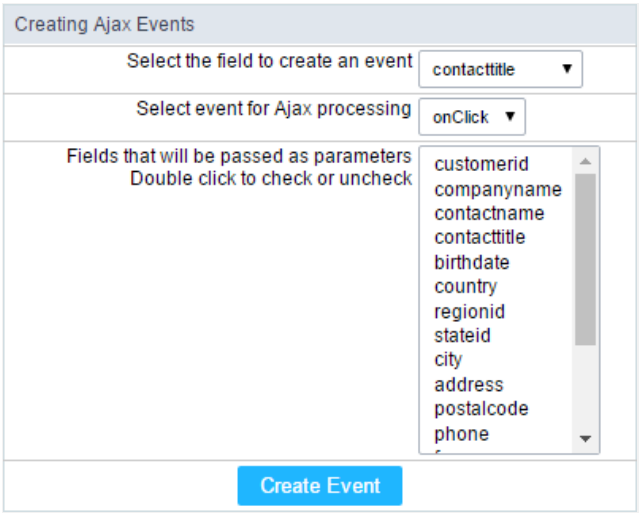
The ajax event OnClick is executed when the field that it's based on is clicked.

- Creating a new ajax event



Creating a new ajax event

- Selecting a field

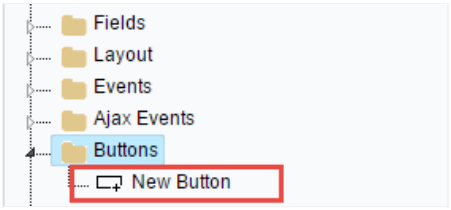


Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

Grid Buttons Settings

In addition to the buttons that comes automatically with the applications, you can also create your own buttons. All the buttons are displayed within the application toolbar.

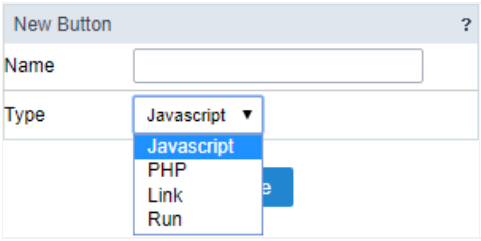


New buttons creation settings

Creating a new button

To create a new button, click on the “New Button” option and enter a name and a button type.

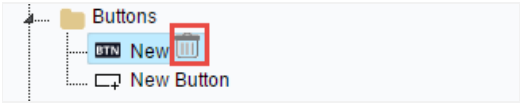
The button Types are: JavaScript, PHP, Link and Run.



Grid buttons type

Deleting a button

To delete a button click on Delete icon (represented by a recycle bin image) in the right of the button name, at the application menu under the Buttons option.



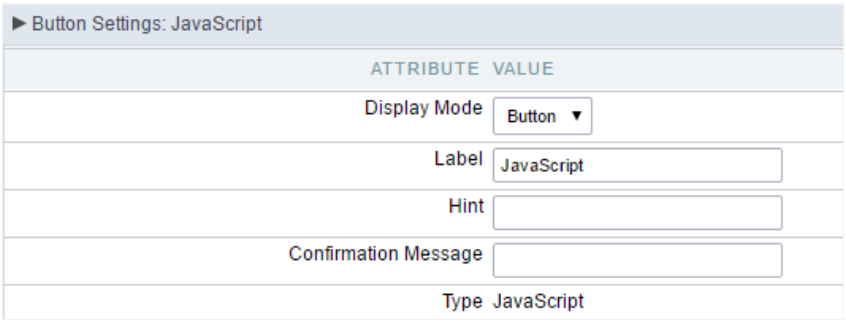
Deleting a button

JavaScript

Display Mode

You can configure the display mode of the javascript button in Image, Button or Link.

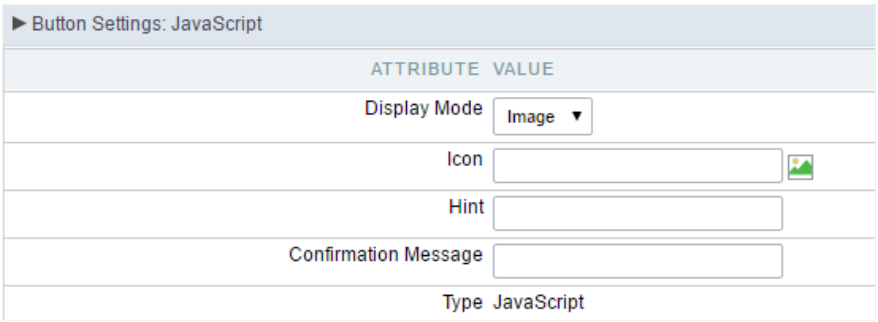
Button



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

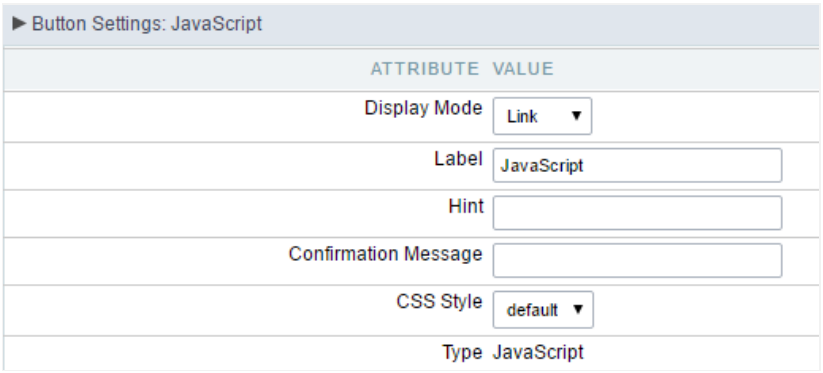
Image



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

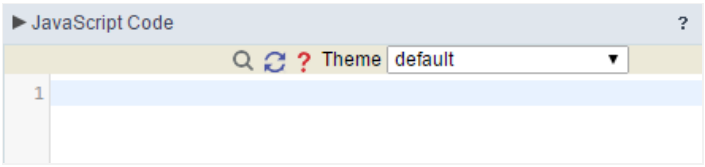
Link



Setting up Javascript Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Code Block



Javascript button coding block.

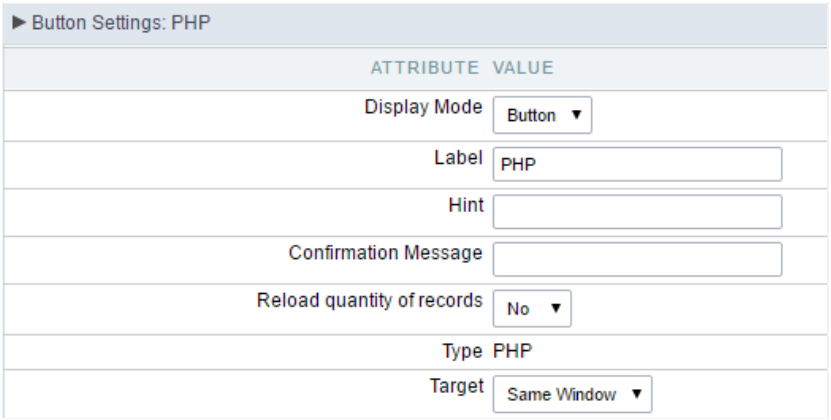
In this block, only JavaScript is accepted.

PHP

Display Mode

You can configure the display mode of the PHP button in Image, Button or Link.

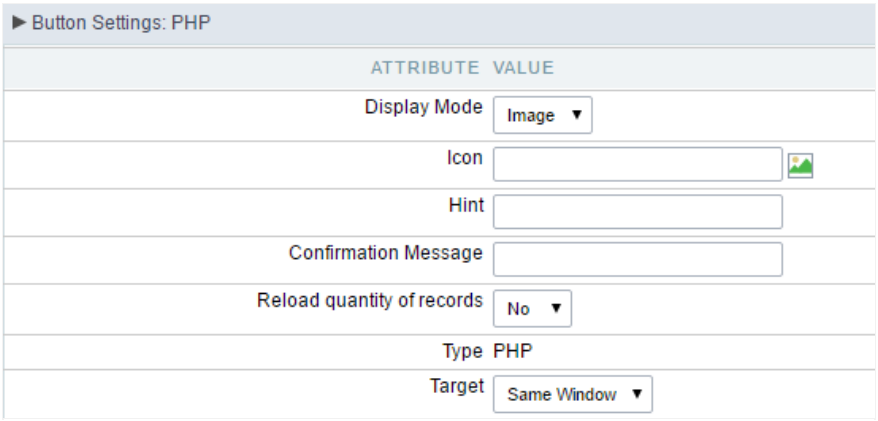
Button



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

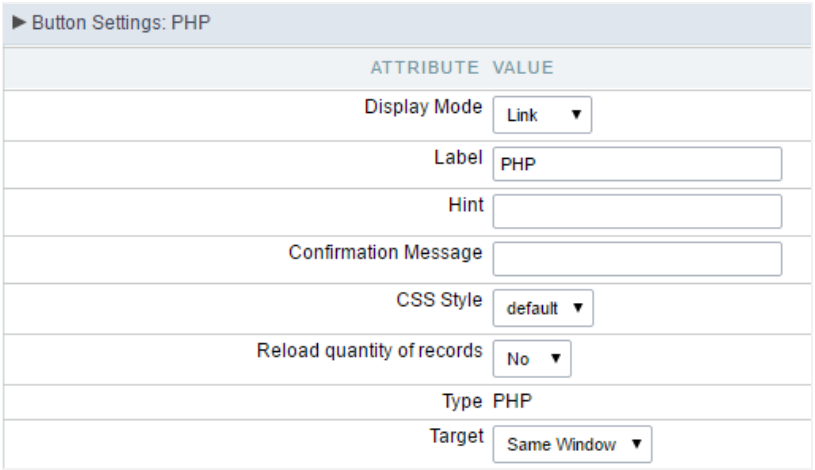
Image



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

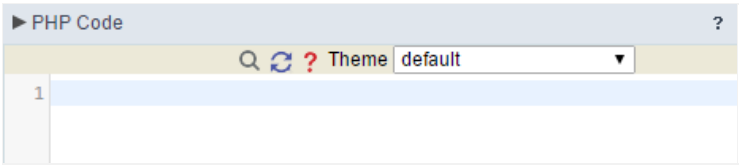
Link



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

Code Block



*Ajax button coding block. *

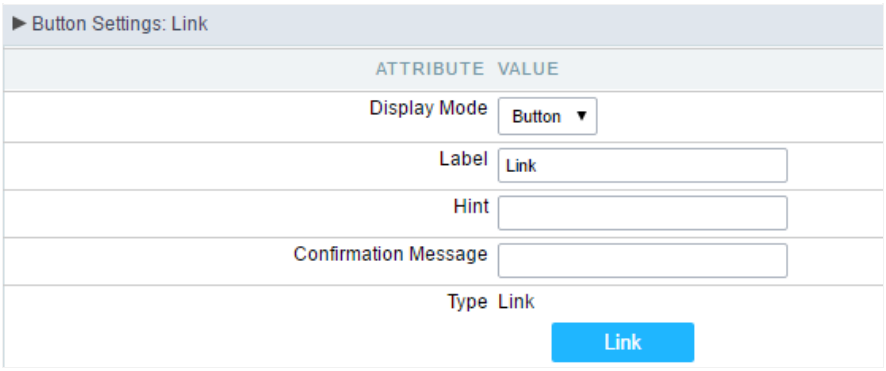
In this block, you can use macros, PHP code and JavaScript.

Link Button

Display Mode

You can configure the display mode of the link button in Image, Button or Link.

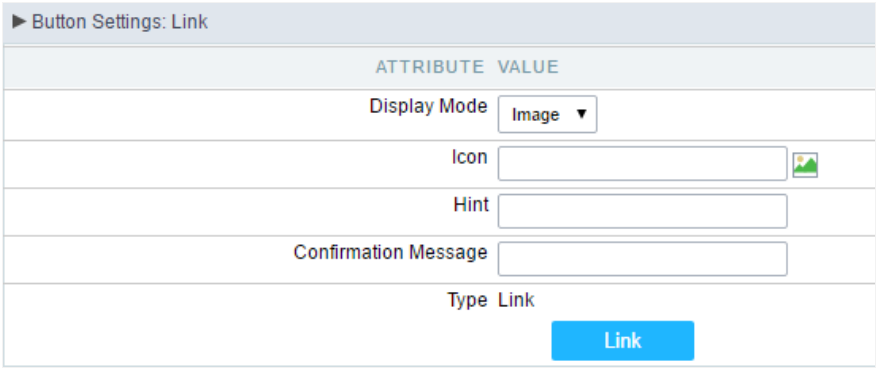
Button



Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Image

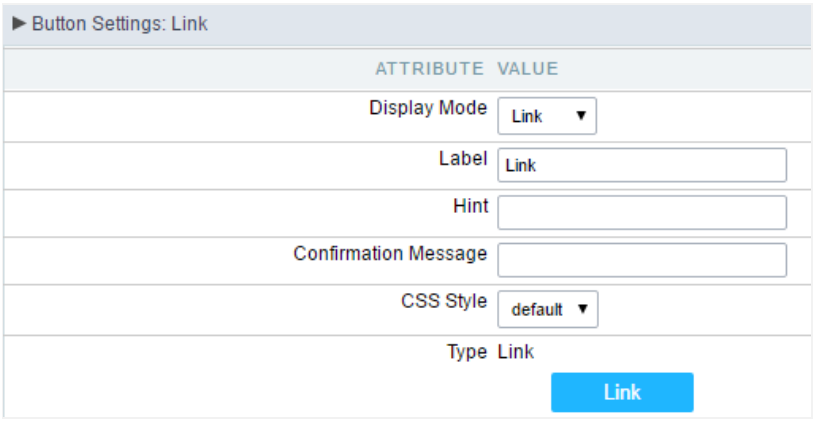


Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.

Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Link

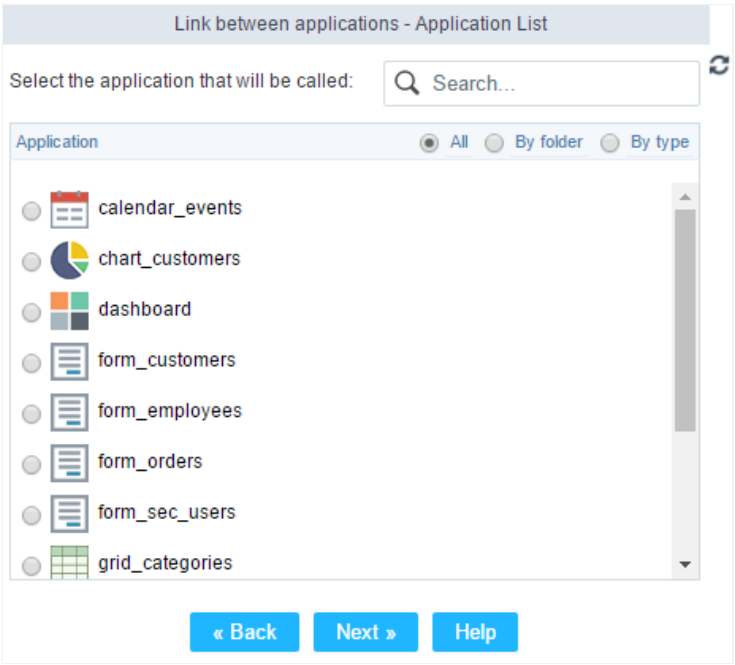


Setting up link Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Setting up the Link

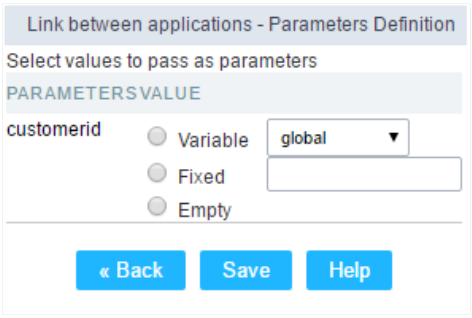
- Selecting the applications



Choosing the application for the button link.

You should select an application to be called from the button link.

- Link Parameters



Choosing the parameters for the button link.

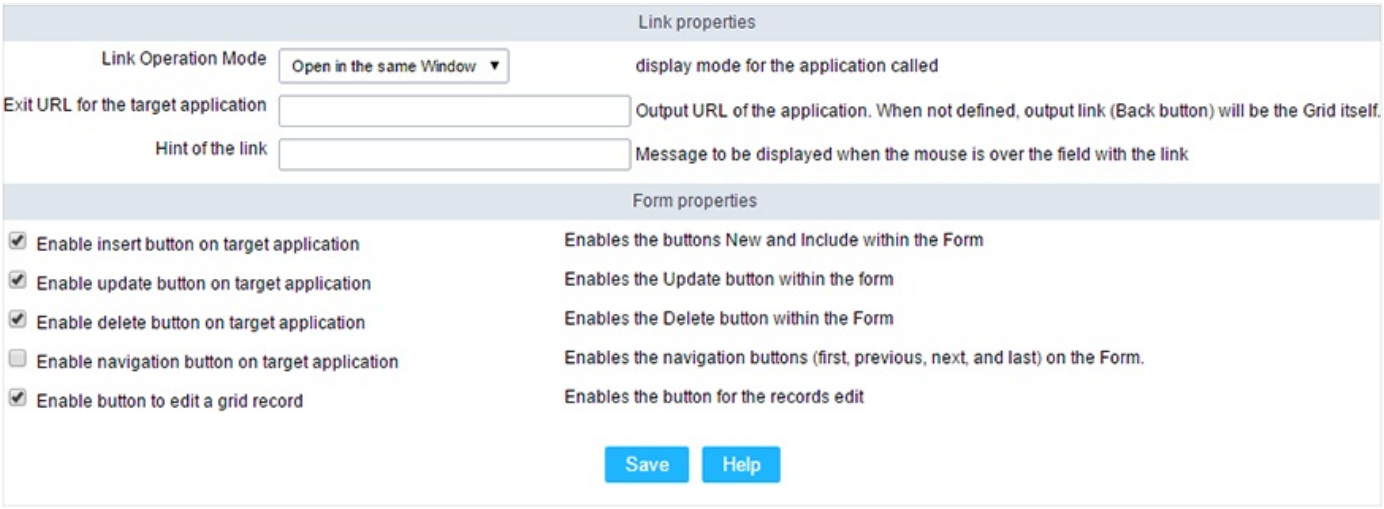
Field Allows you to use an existing field from the current application as a parameter for the link.

Variable Allows you to use a global variable from the current application as a parameter for the link.

Fixed Allows you to inform a fixed value as a parameter for the link.

Empty No value will be passed as a parameter for the link.

- Link Properties (Grid)



Configuring the properties for the link button when the destined application is a Grid.

Link Operation Mode How the link will open (Open in the same Window, Open in another Window, Modal).

Exit URL for the target application URL or an application that be redirect to when exiting the Grid application.

Initial Mode Allows you to define the initial mode of the grid application (Search or Grid).

Number of Lines Allows you to define the amount of lines displayed in the Grid.

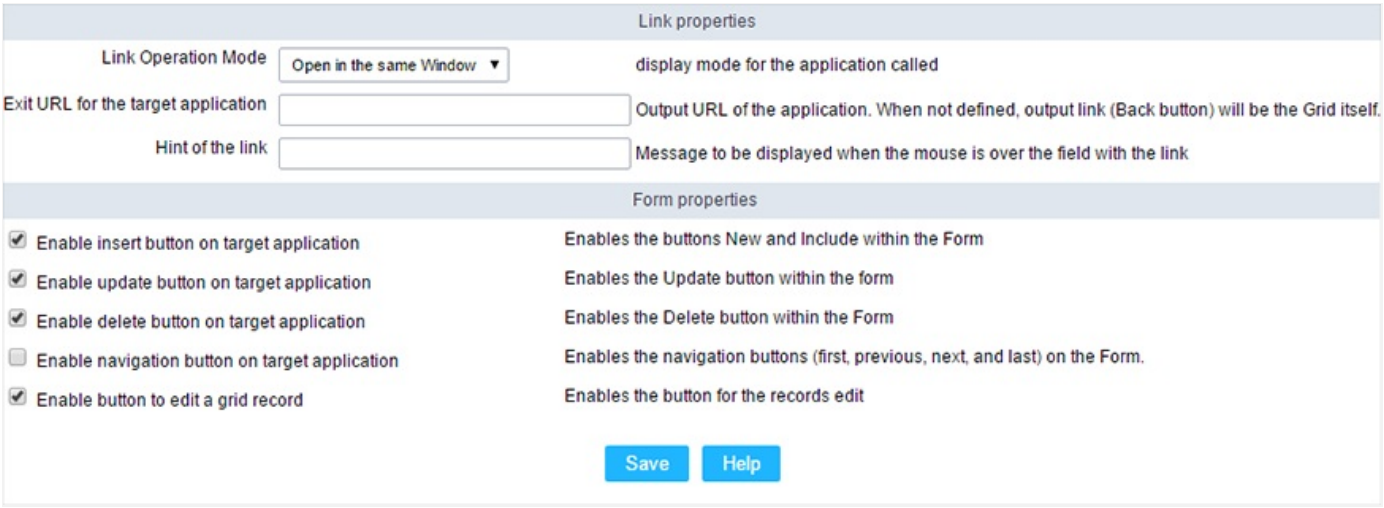
Number of Columns Allows you to define the amount of columns displayed in the Grid.

Paging Enable the paging in the Grid.

Display Header Enable the Grid Header.

Active Navigation Buttons Enable the navigation button (First, Back, Next and Last) in the Grid.

- Link Properties (Form)



Configuring the properties for the link button when the destined application is a Form.

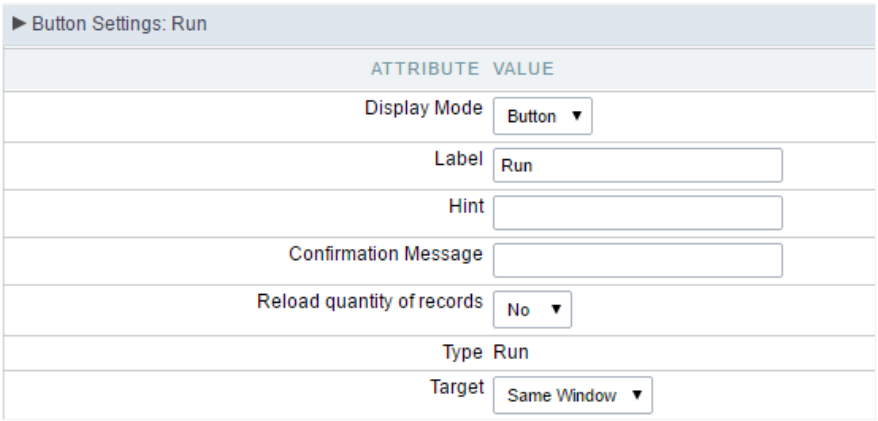
Link Operation Mode	How the link will open (Open in the same Window, Open in another Window, Modal).
Exit URL for the target application	URL or an application that be redirect to when exiting the Form application.
Enable insert button on target application	Enable the “New” button in the Form Application.
Enable update button on target application	Enable the “Update” button in the Form Application.
Enable delete button on target application	Enable the “Delete” button in the Form Application.
Enable navigation button on target application	Enable the navigation button (First, Back, Next and Last) in the Form.
Enable button to edit a grid record	Enable the buttons that allow you to edit the records of a Grid

RUN

Display options

We can configure “Run” button display as Link, Image or Button:

Button

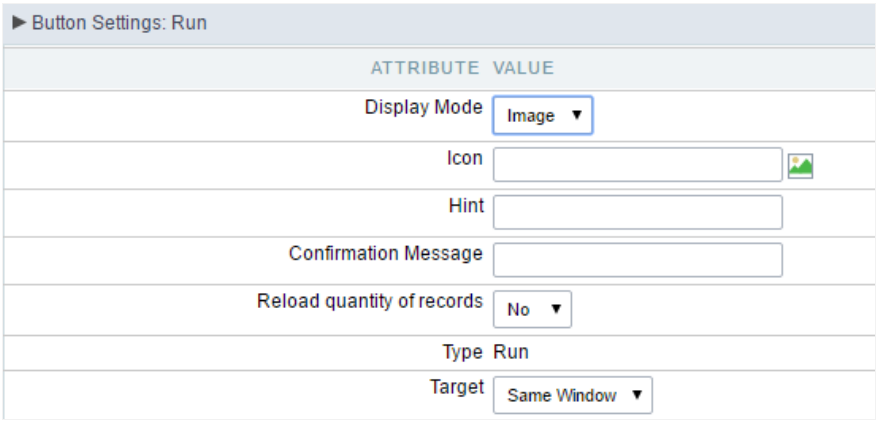


Grid’s Run Button settings - Button type

Display Mode Combo box to select the button display option, you can choose button, image or link.
Label Application button title (text that will be displayed for the button within the application)
Hint Using this option you can set a message for the button hint
Confirmation Message Using this option you can set a confirmation message that will be displayed when the button is pressed. Leave it empty if you don’t need to display any message.

CSS Style	CSS for the the button, if you do not change this option Scriptcase will apply the default application/project CSS. You can customize the buttons' CSS using the option within the main menu "Layout » CSS Buttons"
Reload quantity of records	Option used to update the amount of application records.
Type	Button type description.
Target	This option allows you to set the target window where you will run the button code (the same window, other window, modal)

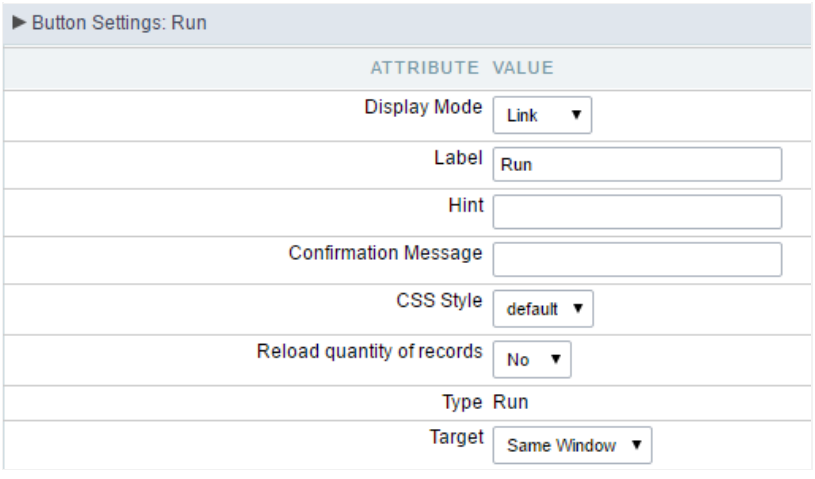
Image



Grid's Run Button settings - Image type

Display Mode	Combo box to select the button display option, you can choose image, button or link.
Icon	Button display icon. You can use this option to select an image (from Scriptcase images or from you computer) to represents the button within the application toolbar.
Hint	Using this option you can set a message for the button hint
Confirmation Message	Using this option you can set a confirmation message that will be displayed when the button is pressed. Leave it empty if you don't need to display any message.
Reload quantity of records	Option used to update the amount of application records.
Tipo	Button type description.
Target	This option allows you to set the target window where you will run the button code (the same window, other window, modal)

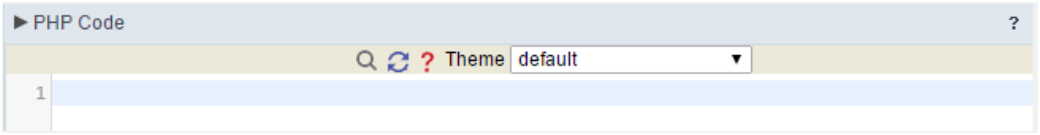
Link



Grid's Run Button settings - Link type

Display Mode	Combo box to select the button display option, you can choose link, button or image.
Label	Text that will be displayed on the button (on running application).
Hint	Application button title (text that will be displayed for the button within the application)
Confirmation Message	By using this option you can set a confirmation message that will be displayed when the button is pressed. Leave it empty if you don't need to display any message.
CSS Style	CSS class name, style created in the theme buttons editor .
Reload quantity of records	Option used to update the amount of application records.
Type	Button type description.
Target	This option allows you to set the target window where you will run the button code (the same window, other window, modal)

Coding Area



Grid's Run Button settings - Coding area

- There are two types of events in the Run button
  - **OnRecord** : Runs after processing on each record selected.
  - **OnFinish** : Runs after processing all records selected.

In this coding area you can use Scriptcase macros, PHP and JavaScript.

Related Links

Related Video

Detail Module

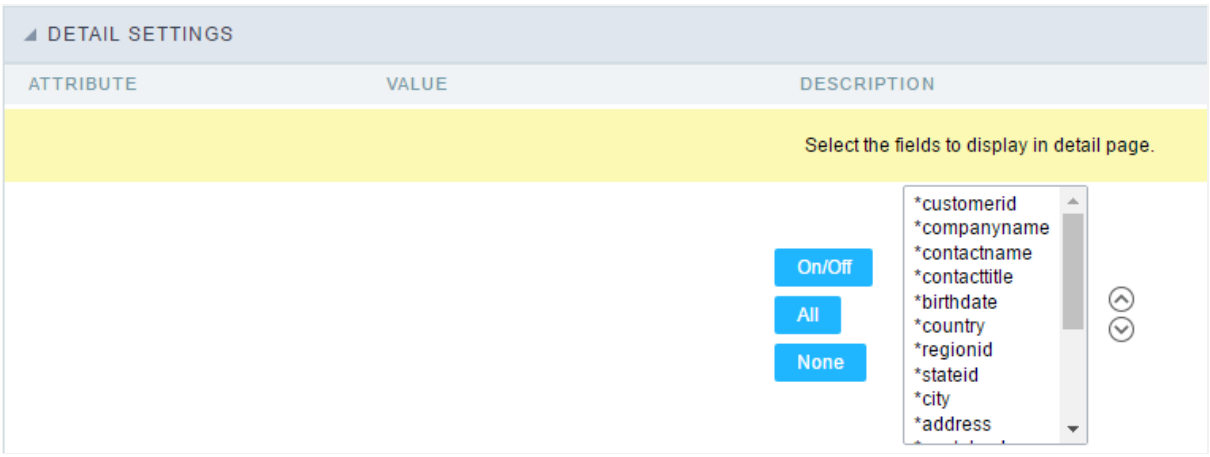
The Grid Detail Module allows you to organize the information displayed according to its significance. It's a complementary application that displays the data in a detailed Form. In order to see the Grid Detail within the Grid the final user needs to click on

 that is displayed for each record.

To enable the Grid Detail Module, access the "Grid Modules" and enable the check box, it uses to come enabled by default:

Detail Settings

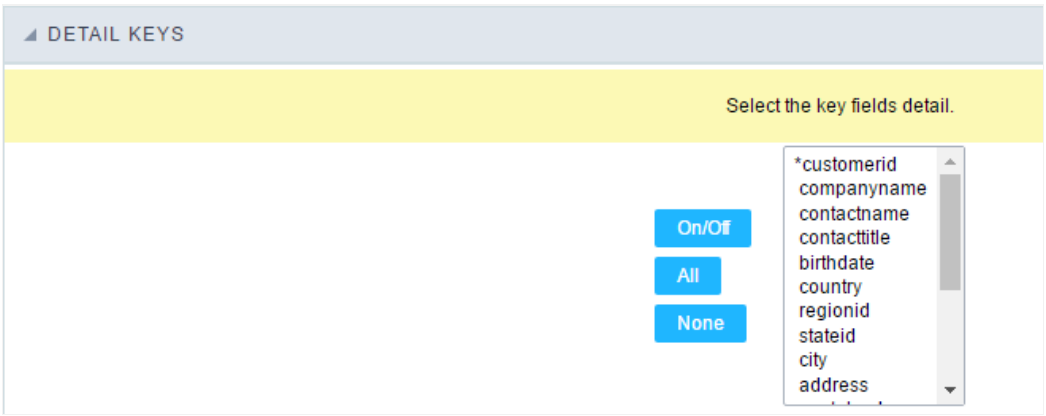
In this section, the user can define how the detail of a particular record should be displayed when it's selected and which registry information should be displayed in this detail.



- **Attributes**
  - **Fields selection for the detail page.** : Allows you to select the desired fields. Just click the field and then click the "On/Off" button to enable/disable the fields.

Detail Keys

Through this interface you can chose the fields that will be part of the "WHERE" clause, it will return the corrected information according to the selected record. ScriptCase uses to identifies the table primary keys, that is being used within the application SQL, automatically. However, sometimes it is necessary to set the primary key manually, especially in cases where the application has a SQL JOIN, involving more than one table.

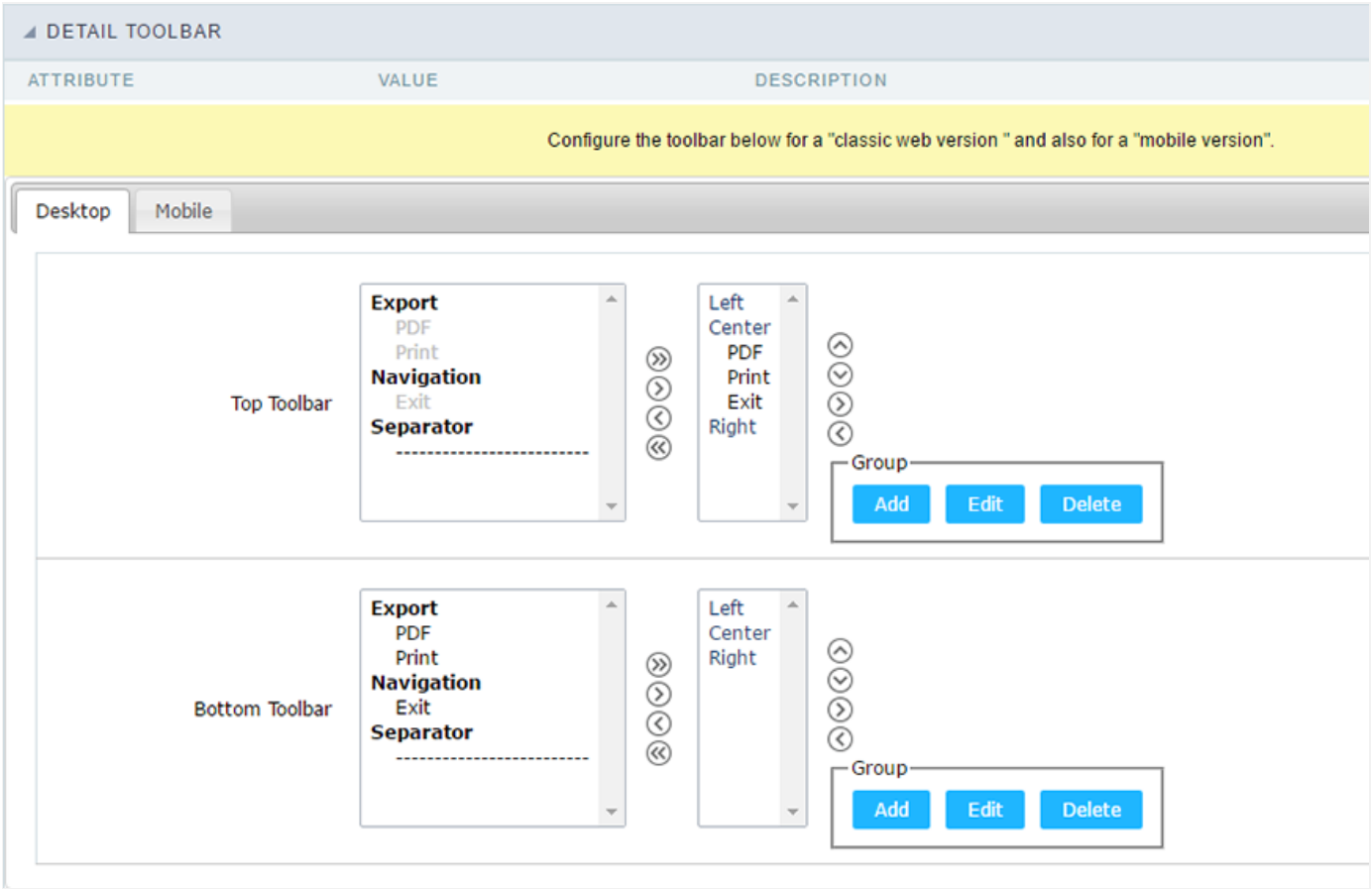


Select here the fields that are keys to the Detail. The application will execute another select command, and you can configure which fields will be passed for this "WHERE" clause:

- **On/Off** : Select the field to be displayed in detail. The selected field (On) gets an asterisk.
- **All** : Check “On” to all fields.
- **None** : Check “Off” to all fields.

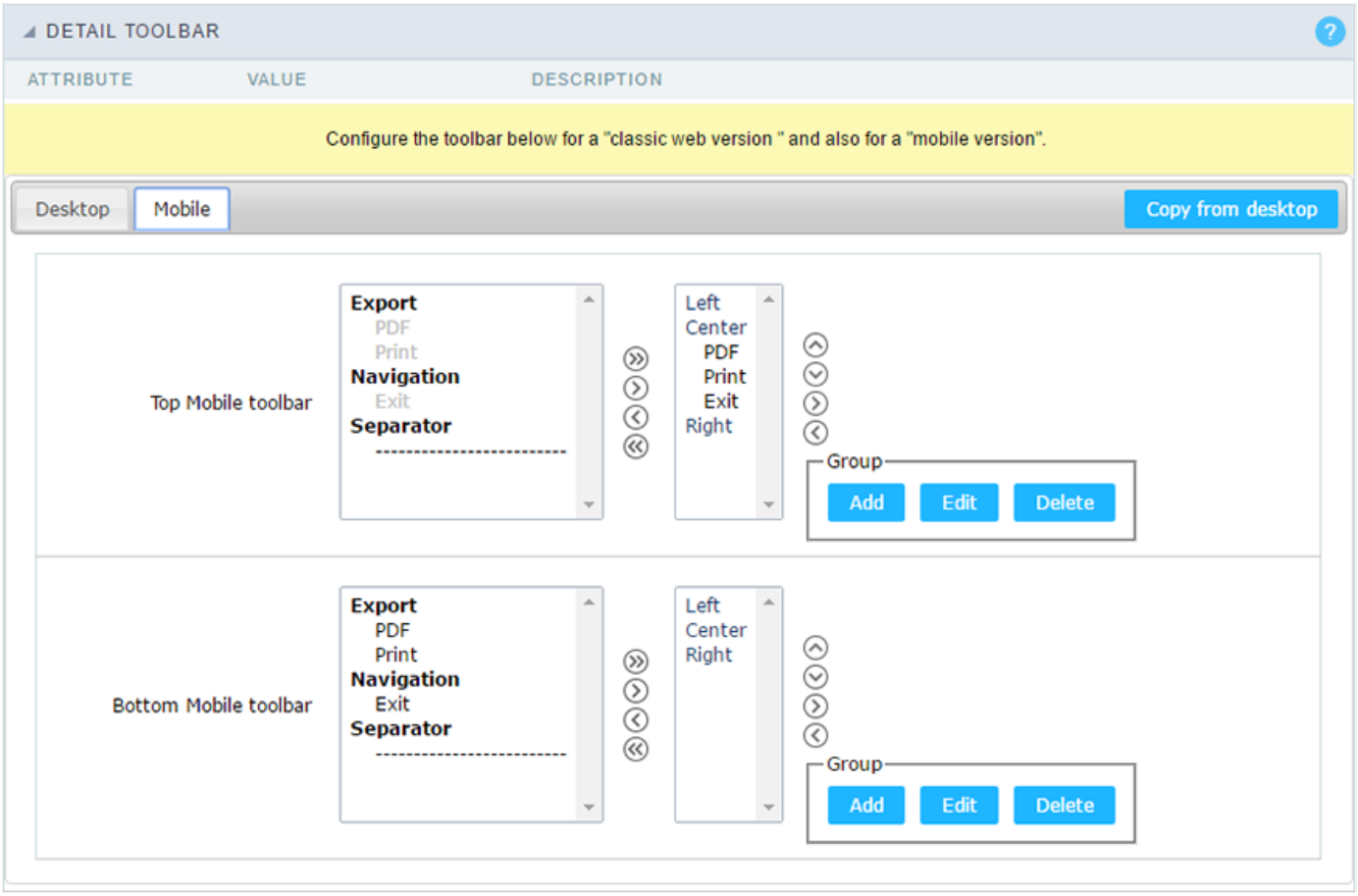
Desktop Toolbar

The application toolbar is divided in two parts: superior and inferior. It is possible to select the buttons for each part independently. A button can even be part of both parts simultaneously.



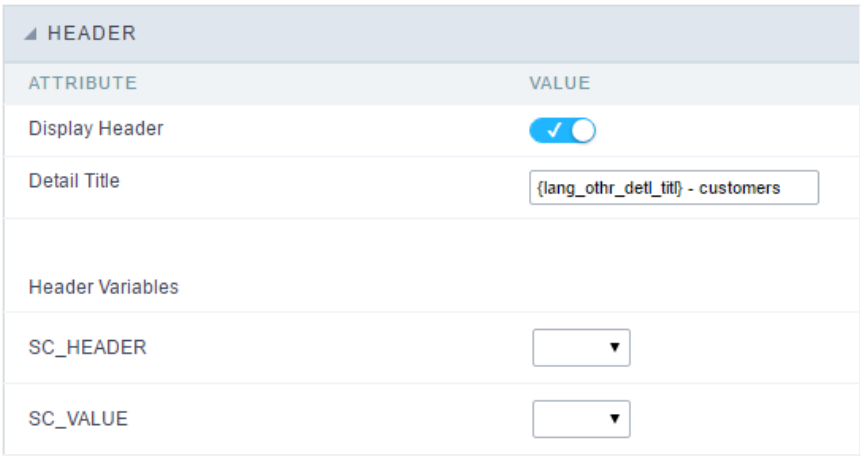
- **Attributes**
  - **Navigation** : Navigation Buttons
    - **Exit** : Ends or return to the previous application depending on how the “Display Detail” is configured. You can change the label within the “Button Settings”.
  - **Exports** : Sets the formats for printing generation:
    - **Print** : Creates a complete report with all Grid Detail data within a HTML printable page.
    - **PDF** : Creates a complete report with all Grid Detail data within a PDF file type.
  - **Othes**
    - **Separator** : Creates a separation line between the buttons for better display, specially for grouped buttons.

Mobile Toolbar



It has the same options of **Desktop** version, by adding only the item “copy of the desktop”, in which, when clicked, performs a copy of toolbar items from **Desktop** to **Mobile**.

Button Settings



Button Displays the buttons available in the application.

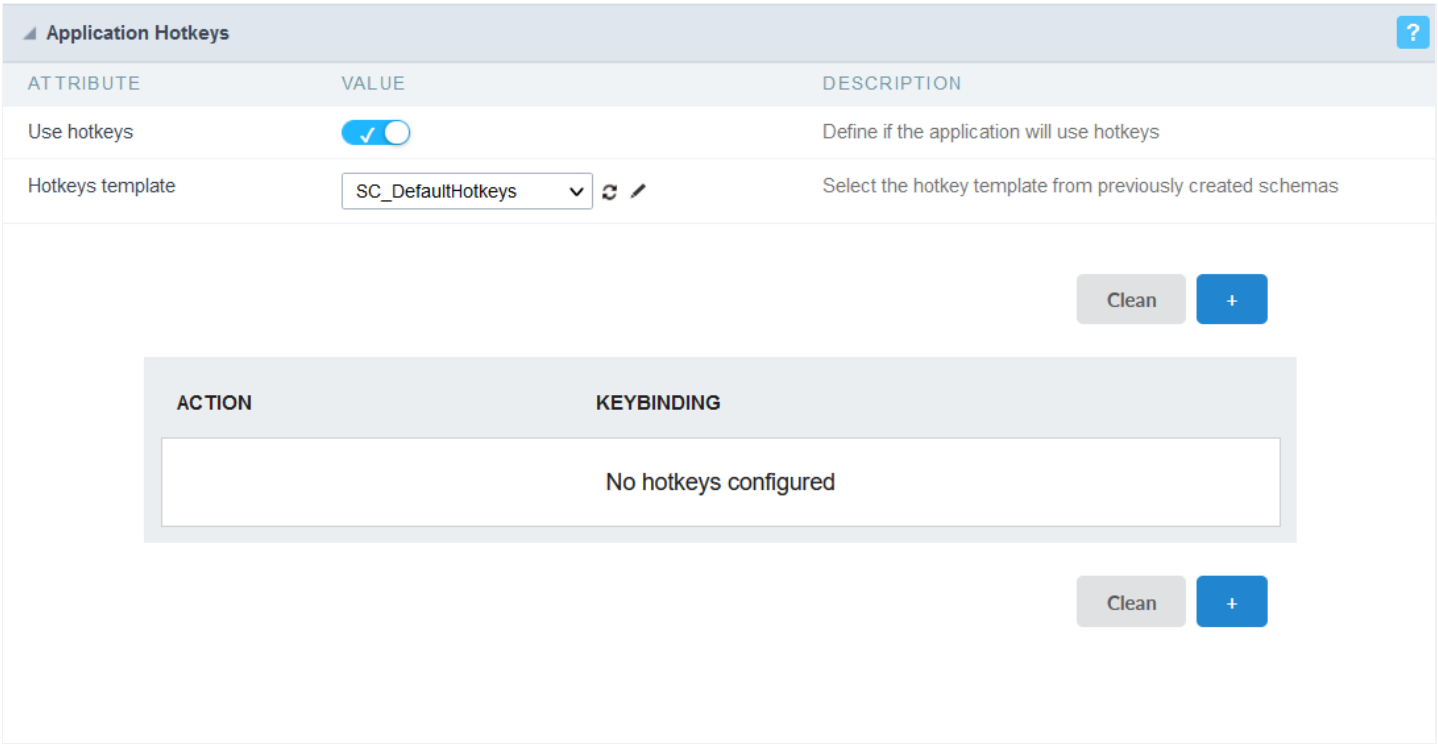
Label Allows you to customize the buttons name from current application. You can also use the option “Application Language” (placed within the main menu “Locales”) to change the button labels for the whole project.

Hint Allows you to set a hint for the buttons. The hint will help the user to understand the button’s action.

Shortcut key Allows you to set a shortcut key to run the button action. Important note: Each browser has its own shortcut combinations and reserved words, you need to check it before implementing this option.

Application Hotkeys

Scriptcase allows creating shortcut keys to your applications. You can select a predefined template or create specific actions for an application.



Use hotkeys

Defines if the application uses hotkeys. When you enable this option, the default shortcut keys settings are disabled.

Hotkeys templat

Select the hotkey template previously created.

Action

Selects the triggered action when pressing the selected key.

Keybinding

Selects the keys responsible for executing the chosen action.

Add “+”

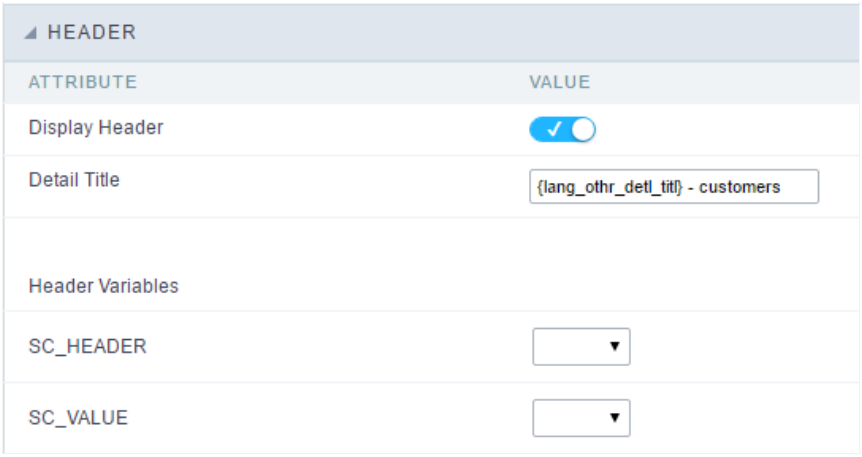
Adds a new action on the keys list.

Clear

It clears the selected hotkeys preference.

Header

In this section is where the content variables that will be part of the header is set.



This screen will change depending on the format chosen header within.

- **Display Header** : This flag determines whether the header is displayed.
- **Detail Title** : Using this option you can set a title for the application, it uses bring a value from the Grid application as default. However it can be customized.
- **Header Variables** : Variable fields can be filled with any of the types displayed in the Combo box. Depending on the type, you must associate a content. Below we describe the existing types:
  - **Field** : When the "__Category" type option is chosen, it will open a combo box next to the fields that are part of the "Select". By choosing one of these fields, you are associating the value of the field to display in the header.
  - **Title** : When this option is selected, it will display the value filled in "**Detail Title**" to the header.
  - **Date** : When you select the "**Date**" type, the system date in the format mm/dd/yyyy will be displayed in the header. There are several display formats using server date and time. The format can be reported in the text field that appears next. Click in the question mark to check out the available formats.
  - **Image** : When the image type field is selected, a field for filling in the image name on the server will appear. To locate the existing images and select one, click on the "**Choose Image**" icon and to make new images available on the server click on "**Download an image**" .
  - **Value** : When "**Value**" type is selected, the content filled in the text field next to that will be displayed in the header, texts and "__ Global variables__" can be entered. Ex: "Name of the Employee: [v_name]".

Nested Grid Overview

The *Nested Grid* functions as a link between query applications and is used to display and maintain supplemental data from the parent application, such as customer order information about a.

Example of application with Grid nested in Iframe

In the example, the nested Grid is set to display below the record, with options to edit and add new items to the order.

Check configuration options

- General Settings
- Creating a nested Grid
- Individual Nested Grid Settings

Nested Grid Settings

On this screen we must define the display configuration of the nested Grids in the main application.

We can define the general behavior of the nested Grids or use the **Define Individually** in the attributes and define the appearance and behavior individually, in each nested Grid created.

Configuration

This configuration block defines how the nested Grid will be displayed in the main application.

iframe mode

This attribute defines whether or not the iframe is used to display the called application.

This implies the full use of the functionality of the query configured as nested Grid, such as: Pagination, filters, export, breaks, columns and ordering.

Title on the same line

This attribute defines whether the nested Grid column titles will be displayed on the same line as the main Application Title.

> This attribute is available when the **iframe mode** and **Enable TreeView** options are set to *No* or *Individually Defined*.

The configuration options are:

- **Yes** - In this option, the title of the nested Grid columns will be embedded in the title of the main application, as in the example below.
- **No** - In this case the **Positioning** will be displayed so that we can define the positioning of the nested Grid.
- **Define individually** - Enables setting the Title on the same line in the configuration of each nested Grid created, so that the behavior can be defined individually.

Example of nested Grids displayed on the same row

Enable TreeView

This option adds a feature to hide/show the nested Grid in the query.

The display of the nested Grid will be defined by the **Positioning**.

> Choosing Yes will disable the Same Line Title option.

Example of Nested Grid with TreeView

Positioning

Defines the location of the nested Grid within the main application.

> This attribute will be displayed when setting *Yes* in the **Title option on the same line**.

The options are:

- **In a column** - A column will be added after the last field of the main application
- **Below a record** - The nested Grid will be displayed below the record line.
- **Define Individually** - Enables the Positioning configuration attribute in the configurations of each nested Grid created, so that the behavior can be defined individually.

Example of nested Grid In a Column with TreeView

Example of nested Grids Under a Record with TreeView

Nested Grid Positioning

This attribute is available if the main application has two or more nested Grids and if they are configured to display **below the record**.

In this case, we can define whether to display **next to each other** or **below each other**.

Example nested grid shown side by side

Sample nested grid shown one below the other

Alignment

Allows you to configure the alignment of the nested Grid in relation to the main query when the Positioning option is Below a Record.

The options are: **Left**, **Right** and **Center**.

> This option is available only if the Location attribute is set to *Below Record*

Remove application padding and margins

Remove margins and padding from the called app to make it fit better on the screen.

Remove

border Removes the borders of the called application’s main table to better fit the display

iframe configuration

> These settings are only available if the **iframemode** attribute in the Settings block is set to *Yes*

Iframe Hight

Allows you to set the iframe height in pixels.

Iframe Width

Allows you to set the iframe width in pixels.

Inicial Module

Defines the application’s initial module on the first load: (Grid, Filter or empty)

Number of Lines

Defines the number of lines to be displayed.

Number of Columns

Defines the number of columns to be displayed.

Pagin

Defines if the records will be shown partially with pagination (navigation), or if it will be total with all records on the same page.

Fix top toolbar

Allows you to pin the toolbar on top.

Fix bottom toolbar

Allows you to pin the toolbar on bottom.

Buttons

Defines which buttons will be available on the nestedgrid toolbar.

Nested Grid Configuration

This configuration block defines whether or not the nested Grid will be exported with the main application.

The **NestedGrid** can be added in the following export types: PDF, [Excel][link_export_excel, XML {: target="_blank"} and JSON

> This setting is only available if the **iframe mode** attribute in the Settings block is set to *No* or *Set Individually*

Each type of export can be defined as follows:

- **Yes** - This option will include the nested Grid when exporting.
- **No** - This option will not include the nested Grid when performing the export.
- **Define individually**: enables the export field in the nested Grids created, so that the display definition in the export is defined individually.

New link

To create a new link with a Nested Grid, click the New Link item in the Links within Nested Grid folder.

Add a new Nested Grid

Enter the name and label of the link to be created.

- **Name** - Defines the name of the Nested Grid. The name field only allows numbers, unaccented letters, and underscores.
- **Label**: defines the display text of the Nested Grid column.
- **Advanced Nested Grid** - defines the use of the iframe to open the Nested Grid, allowing the use of various features of the query

  The **Label** attribute has no character restriction and allows the use of lang.

Application List

All project applications available for the link will be listed. In this step, the developer must select the destination application that will be used in the connection.

Finding Applications

To assist in locating the application, you can use the **grouping buttons** or the **Search** field.

Search field

The search is performed by application name.

Grouping Buttons

In this grouping there are three visualization options:

All

Lists all applications in the project alphabetically **a-z**.

Example of applications without clustering

By folder

Lists applications grouped by their folder, according to the organization of applications in the Explorer Project.

Example of applications grouped by Folder

- **Item 1** - List of folders created in Explorer Project. Select the folder and all applications in the folder will be displayed on the **item 2**.
- **Item 2** - List of applications in the folder selected in the__Item 1__.

By type

Lists applications grouped by their respective type.

Example of applications grouped by type

Item 1 - Types of existing applications in Scriptcase. Select the type of application you want to list, the applications corresponding to the selected type will be listed in **item 2**. **Item 2** - List of applications referring to the type selected in **item 1**.

When selecting the destination application, click on **Next »**.

Definition of parameters

In this step, the developer must inform the values that will be passed to the parameters of the destination application.

These parameters can be:

- Global Variables - Which can be defined in the events or in the target application's SQL.
- Primary Key - For connections created with applications of the types: Form and Calendar.

Parameters

Lists all parameters defined in the target application.

Type

Defines the source type of the value that will be sent to the parameter defined in the destination application.

The options are:

- **Fields** - It uses the value of a field from the source application as a parameter.
- **Variable** - It uses the value of a global variable, defined in the source application, as a parameter. **This option will only be listed if a global variable is defined in an event in the source application**.
- **Fixed value** - It uses a fixed value, defined in the value column, as a parameter. **In this option only alphanumeric values are allowed**.
- **No value** - Using this option no value is passed. When using this option as a parameter for a primary key, the target application will be displayed in include mode.

Value

Defines the value that will be sent as a parameter. The options in this column change according to the selected **Type**.

- When selecting **Campo** - The value column will list all the fields of the source application, which will send the value.
- When selecting **Variable** - All global variables defined in the source application will be listed.
- When selecting **Fixed value** - A field will be displayed for the value to be informed. The use of variables is not allowed in this option and we must use only alphanumeric values.
- When selecting **No Value** - In this case, no value will be sent to the parameter.

Refresh Button

This button enables ajax reloading of the list of parameters, in this way it is possible to change or add a parameter in the destination application without the need to restart the creation of the connection in progress.

Application without Parameter

To show only the records that meet a certain condition, it is necessary to define parameters, for this the query application must have a WHERE clause with a global variable configured in its SQL.

WHERE grid definition example

Editing nested grid

In the application menu inside the Subqueries folder there is a links folder that contains all the subqueries called by this application. Clicking on a Nested grid will bring up a screen for editing.

General Settings

In this settings block we can define the Label of the Nested grid and change the Connection settings, such as adding or removing parameters. It is also possible to change some attributes defined as **Define Individually** in the general configuration, in the **Subqueries > Configuration** menu.

> [Click here][link_conf_ligacao_subquery]{:target="_blank"} to see more details about the link configuration in the Nested grid.

- **Label** - defines the display text of the Nested grid column. This attribute has no character restriction and allows the use of lang.
- **Ligação** : This attribute shows data about the current link with the Nested grid, showing the application and the parameters used. To change any attribute of the link, click the Edit Link button.
- **Iframe Mode**: Defines whether or not to use the iframe to display the Nested grid. This attribute is displayed in the Nested grid configuration if it has been configured as **Define Individually** in the general configuration in the **Subqueries > Configuration** menu.
- **Positioning** - Defines the positioning of the current Nested grid. This attribute is displayed in the Nested grid configuration if it has been configured as **Define Individually** in the general configuration in the **Subqueries > Configuration** menu.
- **Enable TreeView** - This option adds a function to hide/show the Nested grid in the query. This attribute is displayed in the Nested grid configuration if it has been configured as **Define Individually** in the general configuration in the **Subqueries > Configuration** menu.

> ## iframe setup
>
> This configuration block is displayed only when the **iframe_mode** attribute is enabled.

- **Display** : This attribute allows you to define which elements of the Nested grid, if the Nested grid uses any of these resources, will be displayed in the main query.

The **IframeMode**, **Positioning** and **Enable TreeView** attributes are displayed in the

> The **Label** attribute has no character restriction and allows the use of lang.

Edit view

This block displays the attributes defined as **Define Individually** in the *Subquery Configuration* configuration block in the **Subqueries > Configuration** menu.

- **Display** - Defines if the *Label of the field*,* the number of lines* and the *total* of the Nested grid will be displayed.
- **PDF** - defines if the Nested grid will be displayed in the PDF export.
- **Excel** - Defines if the Nested grid will be displayed in the Excel export.
- **XML** - defines if the Nested grid will be shown in the XML export.
- **JSON** - Defines if the Nested grid will be displayed in the JSON export.
- **Theme**: When checked, forces the display schema of the Nested grid to be the same as the main query.

Display options in Nested grid

- **1** : Display field label.
- **2**: Sequential display (Line number).
- **3** : Show total.

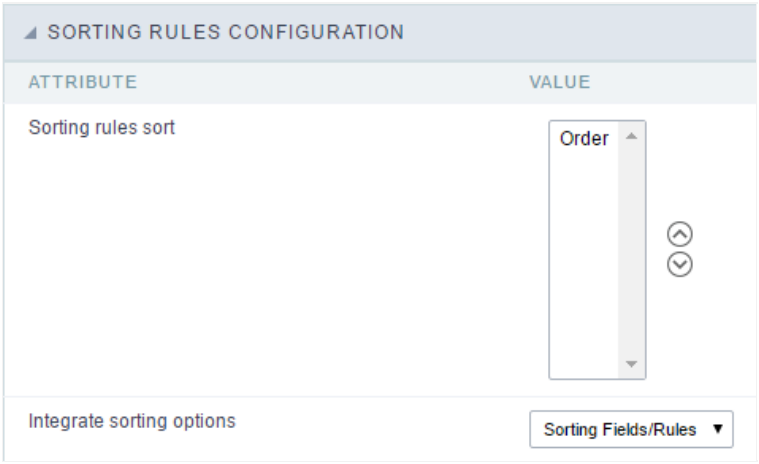
See configuration

This configuration block is displayed with the following configuration:

Attribute	Value
Title on the same line	Not
Enable tree view	Not
Positioning	in a column

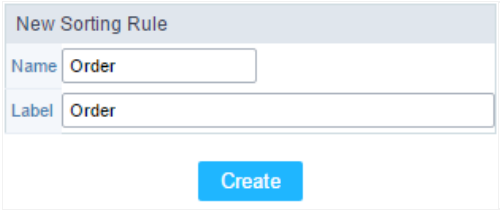
- **Horizontal Alignment**: Defines the horizontal alignment of the Nested grid. It can be left, right, or center aligned.
- **Vertical Alignment**: Defines the vertical alignment of the Nested grid. It can be aligned at the top, middle or bottom.
- **Background Color**: You can set the background color. If not completed, the current display scheme query applications background color will be used.
- **Horizontal Alignment of the Title** : Defines the horizontal alignment of the Label of the Nested grid. It can be left, right, or center aligned.
- **Title vertical alignment**: Defines the vertical alignment of the Nested grid label. It can be aligned at the top, middle or bottom.
- **Bold** : Sets the title (label) of the Nested grid to be bold.

Sorting Rules



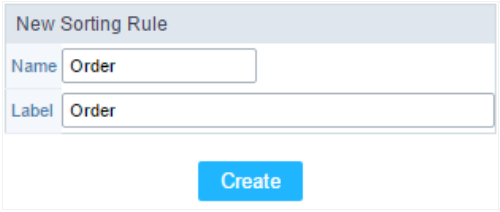
- **Attributes**
  - **Sorting rules sorting** : Allows you to change the sort display of the sorting rules. To configure this option, an icon is displayed (set up ordering) that when clicked, allows to select the field and your display order ascending (ASC) or descending (DESC) .
  - **Integrate sorting rules** : This option allows you to set whether to use the sorting rules together with the regular sorting options or not. You can choose the display options between:
  - **Sorting Fields/Rules(default)** : This option allows you to apply the rules that have been created together with the regular fields sorting (descending or ascending)
  - **Sorting Fields** : This option allows you to apply the fields sorting only, discarding the sorting rules (descending or ascending)
  - **Sorting Rules** : This option allows you to apply the sorting only according to the rules that have been created, discarding the fields sorting.

Creating a Sorting Rule



- **Attribute**
  - **Name** : Field to inform the new rule name.
  - **Label** : Field to inform the name that will be displayed in the application.

Setting up a sorting Rule

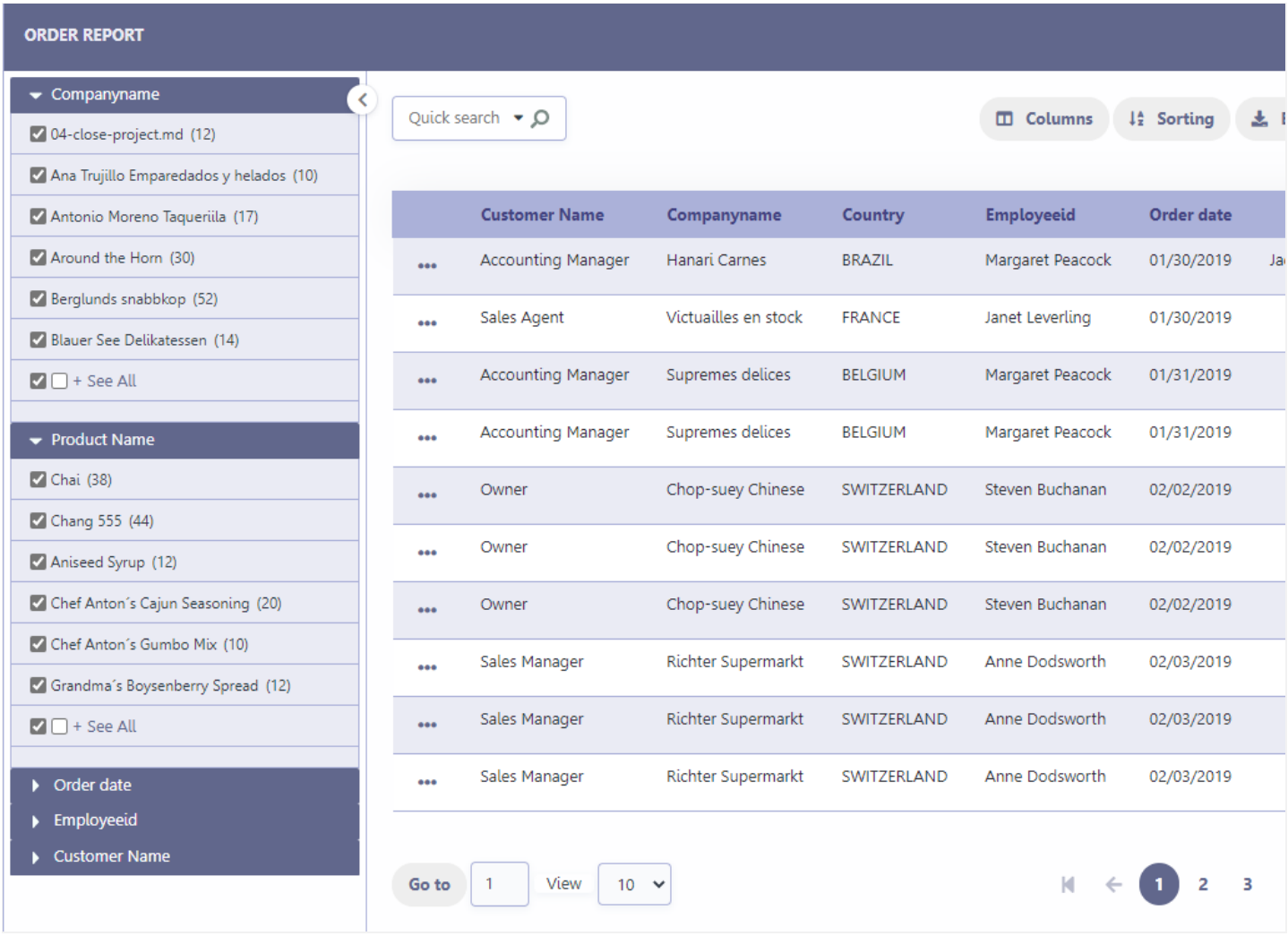


- **Attributes**

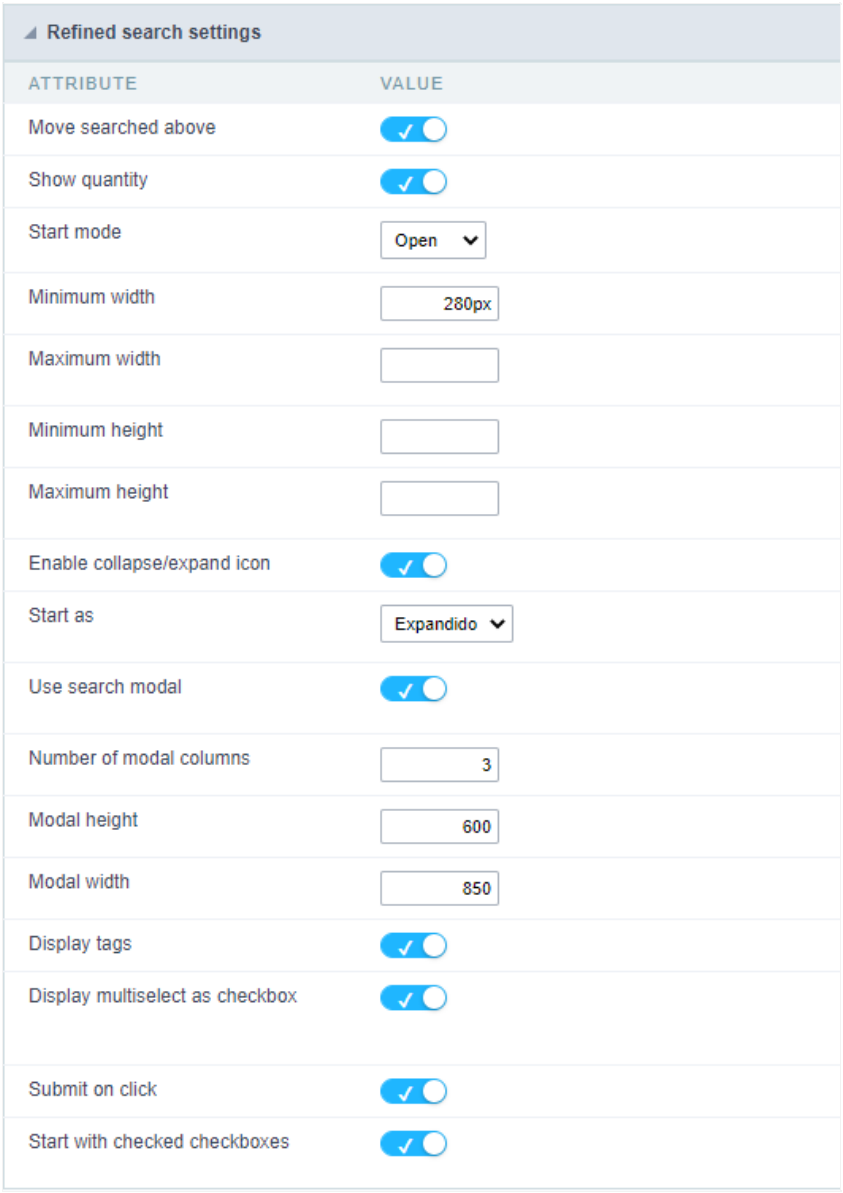
- **Label** : This option allows you to enter a name that will be displayed when the application is executed.
- **Select the fields for a sorting rule** : Allows you to select through the selection bar which fields will be displayed (fields that are in the frame at the right side) and the fields will not be displayed (fields that are in the left side frame). To sort the fields the way you want, use the sort bar located on the right side of the frame. To apply the sorting type to the field, simply select the straight side frame field and select the Ascending mode, where displays a “+” or a “-“ by the field side to indicate if the sort order is Descending.

Refined Search

The refined search is a feature where you can integrate a search interface, next to the Grid, limiting values according to a universe that exists in the connected database.



Settings



Move searched above

When you filter by a field, this field will be moved to the top of the refined search stack.

Show quantity

Alongside the values of the fields, you will see the existing amount. Ex.: Brazil (1547)

Start mode

This option sets whether the fields select options will start open or closed.

Minimum width

This option sets the minimum width for the refined search fields, value in pixels.

Maximum width

This option sets the maximum width for the refined search fields, value in pixels.

Minimum height

This option sets the minimum height for the refined search fields results, value in pixels.

Maximum height

This option sets the maximum height for the refined search fields results, value in pixels.

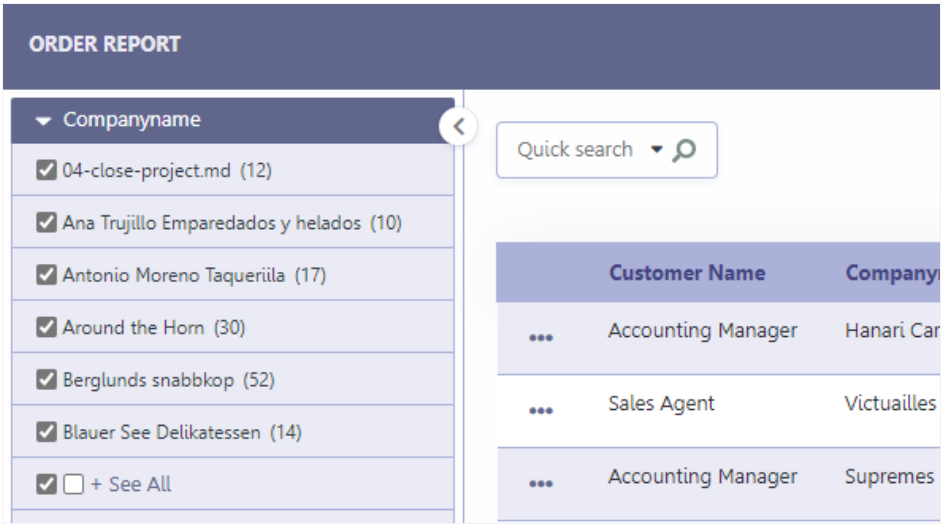
Collected/Expanded

Enables the option collapse or expand the **refined filter** options panel to improve the Grid visualization.

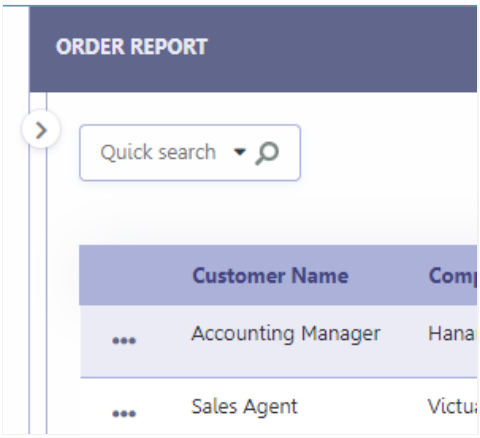
Start as Collapsed/Expanded

Available only when **Collapsed/Expanded** option is active, dfines whether the refined filter will start open or closed.

These are the options:



Start open



Start closed

By default, for new applications, this option will come with the value

Start Open

Use modal

When enabled, sets the modal opening when clicking the **View All** option.

*The button **+ View All** is defined in the filter filds settings, by limiting the amount of records displayed in the field. For more details Click Here.*

Example of modal screen

Number of modal columns

Defines the number of columns for displaying records in the modal.

Only the number must be informed, without measurement unit, otherwise the value will not be considered.

Modal width

Defines the pixel width of the modal that will be displayed.

Only the number must be informed, without measurement unit, otherwise the value will not be considered.

Showing tags

Defines whether or not to display tags with selected filter values.

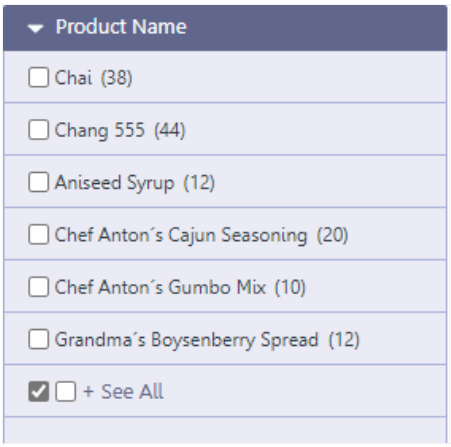
When active the tags are displayed, as in the image below.

Display as checkbox

Defines whether filter values will be displayed with checkboxes or as links.

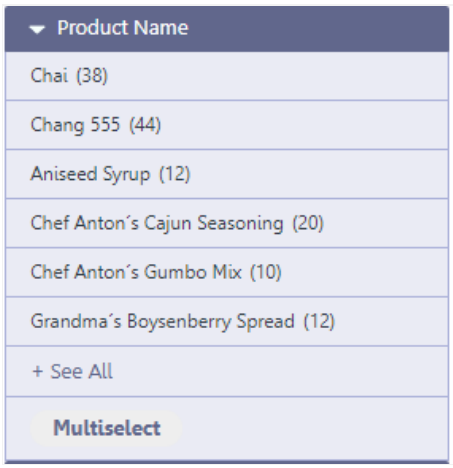
Example of display with checkbox

When enabled, the values will have a checkbox indicating your selection whether they are selected or not.



Display example without the checkbox

When disabled, filter values are displayed as a link, and the select button is shown so that more than one value can be selected.



Submit when scoring

When active, the filter is applied instantly after selecting the desired value.

By activating this option, the application is reloaded with each selection of filter values.*

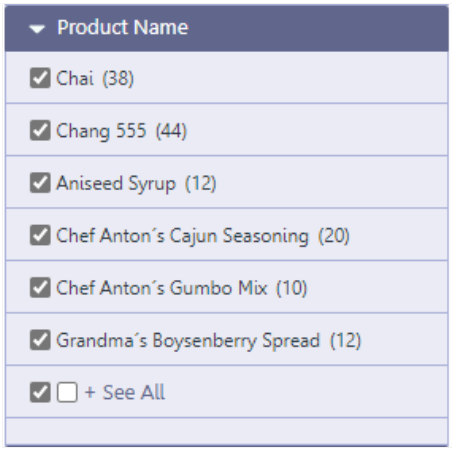
Show marked checkbox

Defines the initial format of the filter values checkbox.

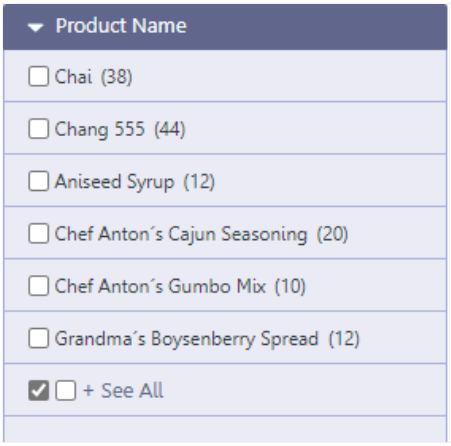
When active, when accessing the application, the filter checkbox is already marked, representing the application's data universe. This way, the user will have to uncheck the options that he does not want to see.

When disabled, when accessing the application the filter checkbox will be unchecked. This way, the user will have to check the options he wants to filter.

Example of enabled option



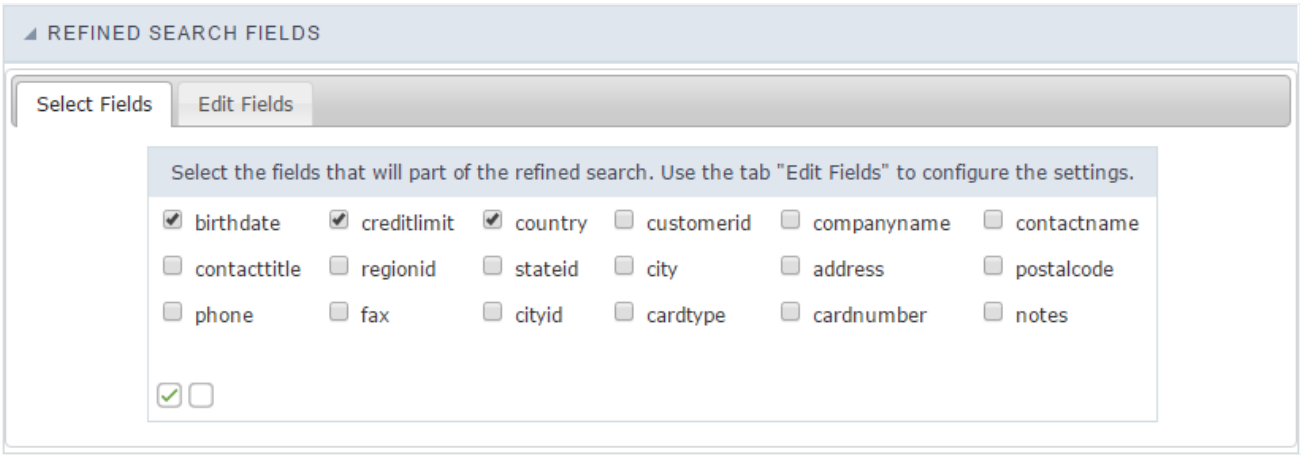
Example of disabled option



Select fields

Refined Search fields

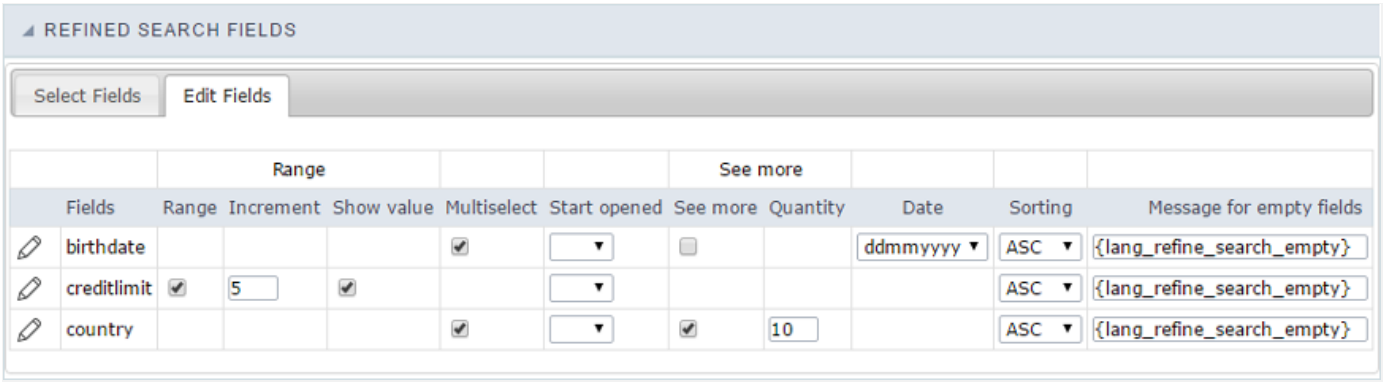
The refined search is automatically added to the Grid application when one or more Grid fields are selected in the tab “select fields”.



Selecting fields of refined search

Edit fields

In the tab “Edit fields”, you can configure each field according to what you need to display within the application.



Fields

Listing of selected fields in the tab “select fields”.

RangeThis

option activate a slider with intervals between the smallest and largest value.

Increment

When using the range this option will set an integer value to increase the range.

Show value

When checked, displays the range values in the slider.

Multi select

Enables multiple selection values to perform the filter.

Start opened

This option sets the initial mode for fields selection display in refined search.

See more

When checked, enables the option to “see more” for the filter records. This is recommended when the selected field has a big amount of records. You will be able to click to “see more” or “see less”.

Quantity

This option sets the initial amount of records to be displayed for each field selected.

Date

This option sets the date format that is displayed in the date fields inside the filter.

Sorting

This option sets the records sorting for each selected field.

Message for empty fields

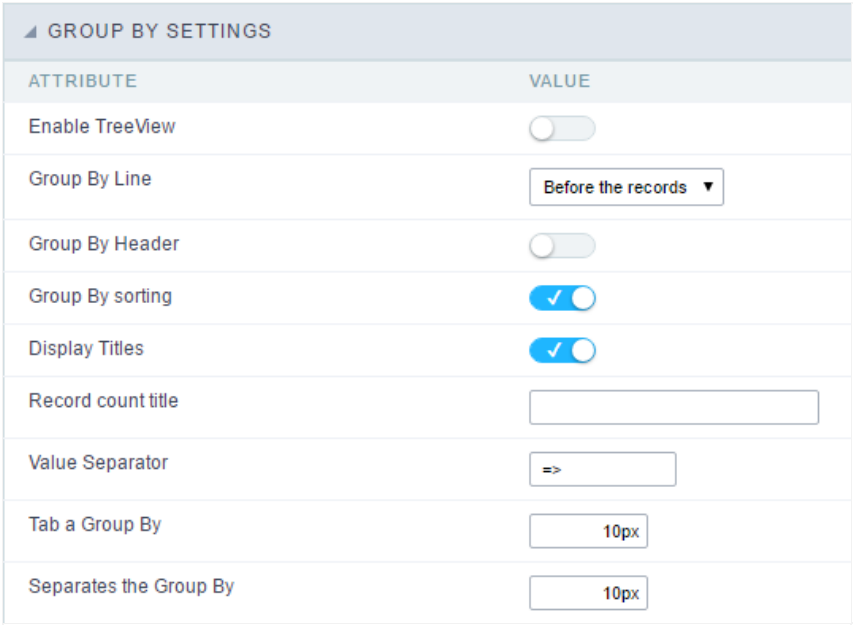
This option sets a message to be displayed when the field has empty values. You can also use a variable from Scriptcase language

system.

Group By

Group By Settings

These are the general settings of the “Group By” in the Grid, independently of it’s type, static or dynamic. Here we can define some Group By viewing options, such as the use of Tree view, for example.



The option **Enable TreeView** isn’t available in the Infinite Scroll.

- **Enable TreeView** : Defines the use of TreeView in “Group By”, which allows you to expand or collapse the records displayed in Group by.
- **Group By Line** : Sets the positioning of group by in relation to group records. The options are **Before the records** or **After the records**.
- **Group By Header** : Lets you display the Group By header on all pages.
- **Group By sorting** : It allows to order the fields respecting the Group By.
- **Display Titles** : Allows the label display of the fields within the groups.
- **Record count title** : Sets a title for the records quantity column.
- **Value Separator** : Define the separator between the title and the field value in the GROUP BY.
- **Tab a Group By** : Sets the left margin of Group By. The value must be informed in pixel.
- **Separates the Group By** : Sets the spacing between two Group By. The value must be informed in pixel.

Dynamic Group By

In this screen we define the fields available in Dynamic Group By and will be summarized in Grid and Summary when the Dynamic Group By is used.

Select Fields

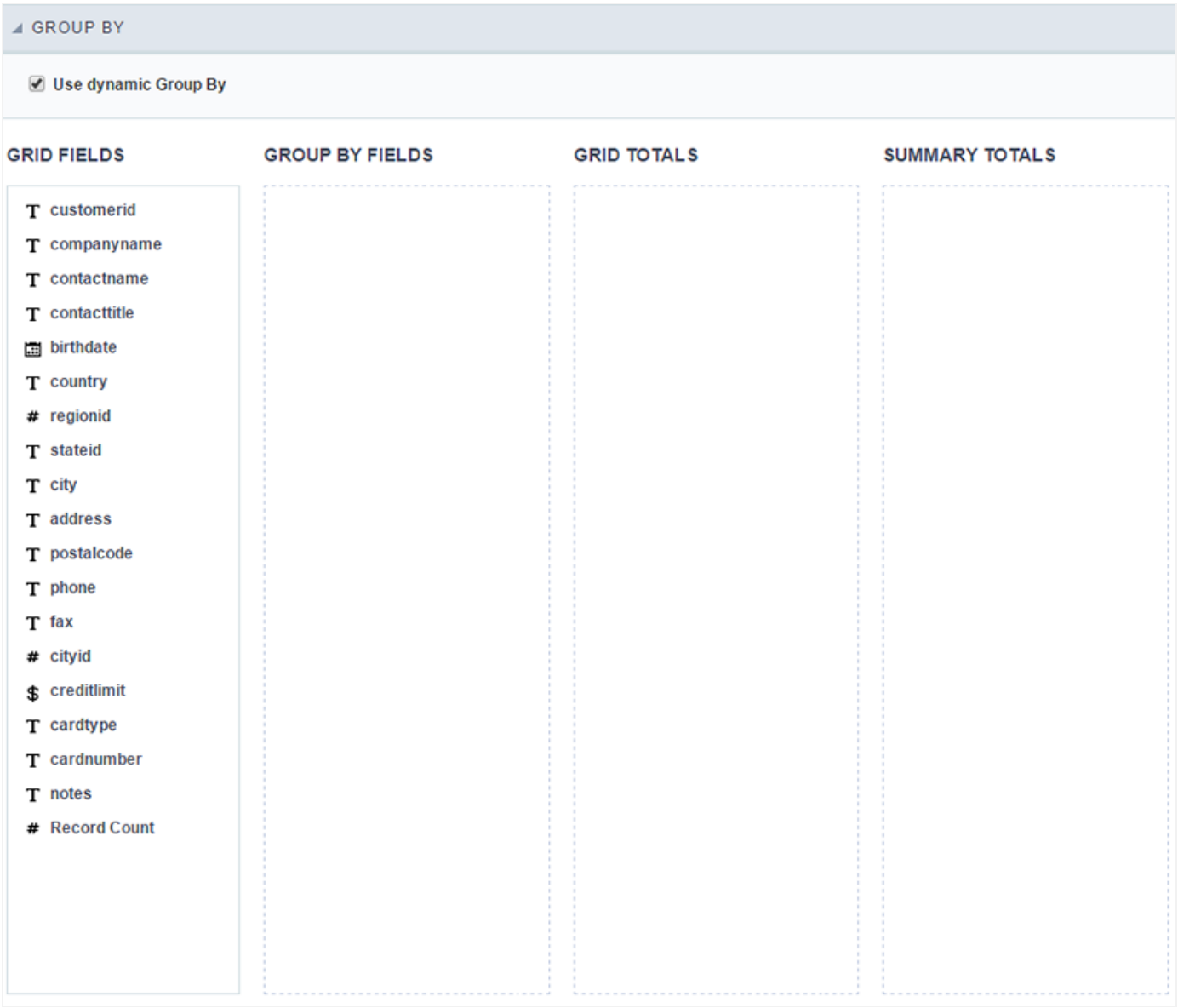
In this screen we define the fields available in the Dynamic Group By and will be summed in the Grid and Summary when the dynamic Group By is used.

This Group By works independently from any other Group By configuration or Totals already performed, that is, the fields defined to be summarized in the Grid or Summary in the Dynamic Group will be visible only when the Dynamic Group is used.

We will be able to see all available fields for the Group By configuration and totals in the **Grid Fields**.

The **Use Dynamic Group By** option enables the Group By in the run-time application.

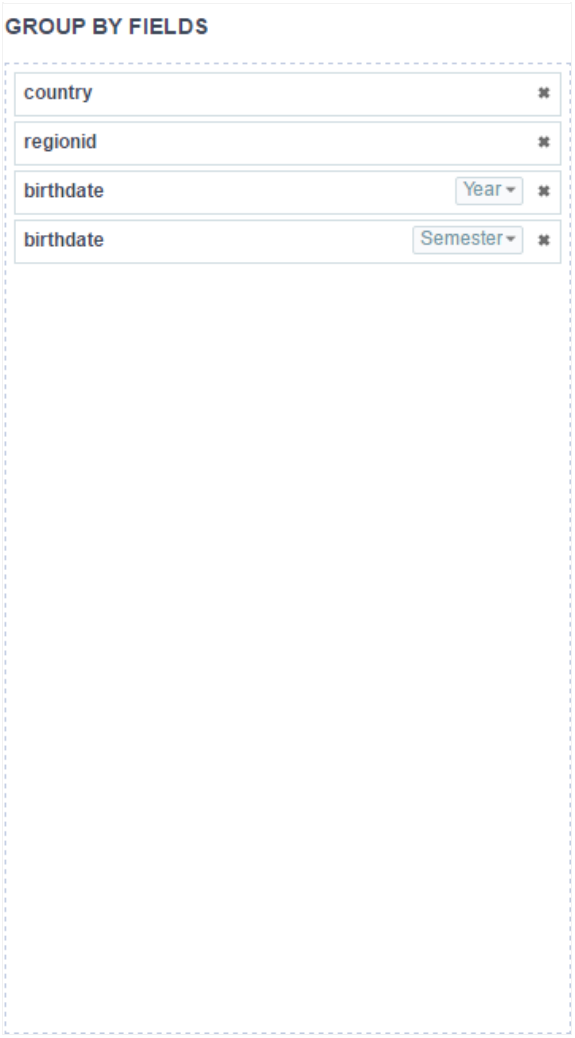
Fields used when using the Grid must be configured to be displayed on the Grid.



- **Grid Fields**: List of all the fields of the application.
- **Group By Fields**: Defines the fields that are part of the Group By.
- **Grid Totals**: Defines the fields that are part of the Grid Totals.
- **Summary Totals**: Defines the fields that are part of the Summary Totals.

Group By Fields

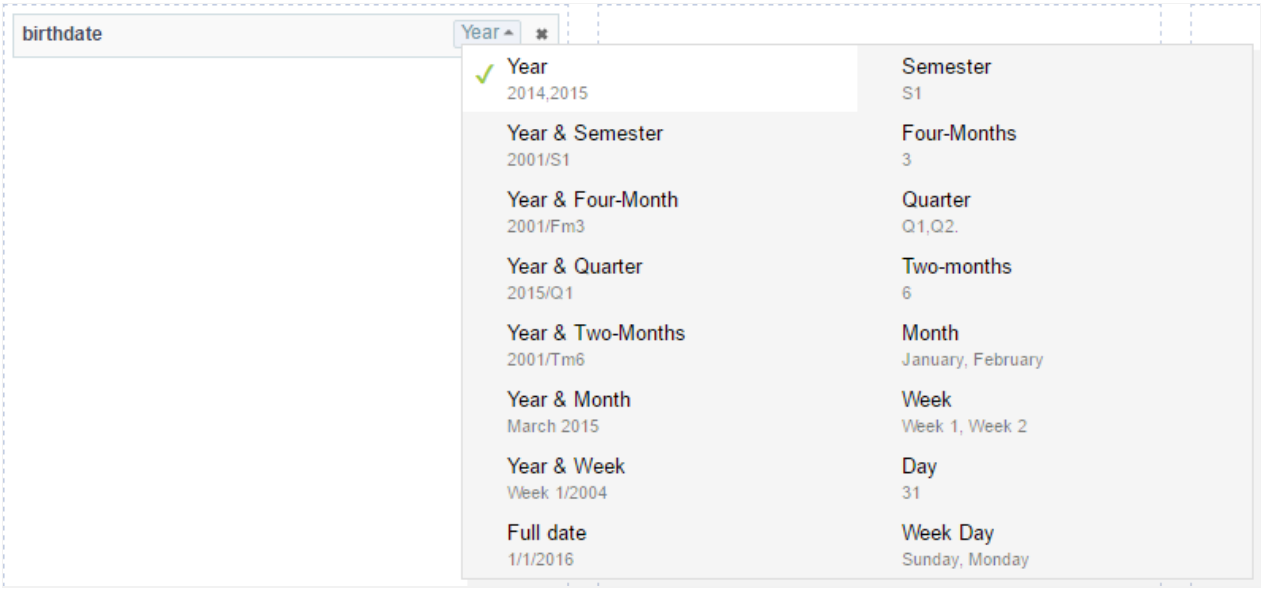
We must drag and drop the fields that will be part of the Group By in **Group By Fields**.



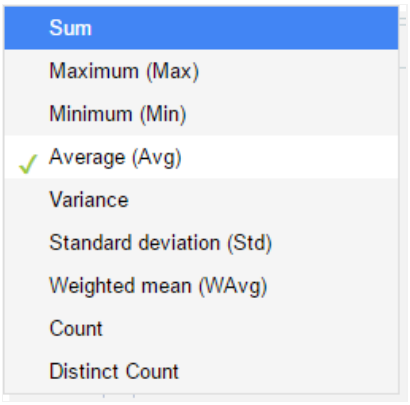
Each field can only be added once to the **Group By Fields**, except for the date and datetime fields.

Date and Datetime fields

For date and datetime fields, some display intervals have been added, so these fields can be added two or more times to the Group By.



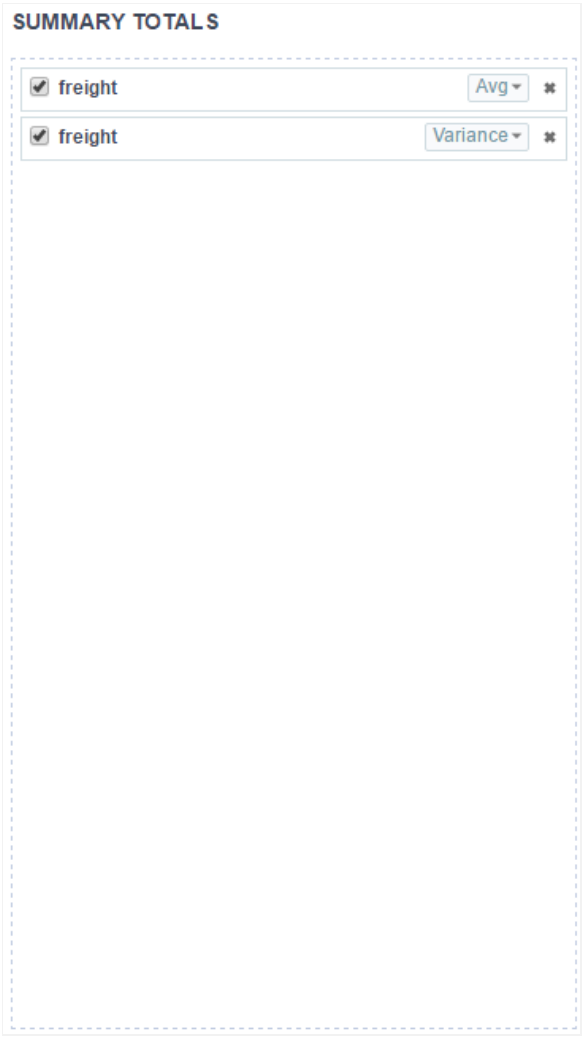
All Grid fields can be added to the “totals”, the only difference being the functions available. Numeric fields can use all the available summarization functions, since the non-numeric fields can use only the count and the different count.



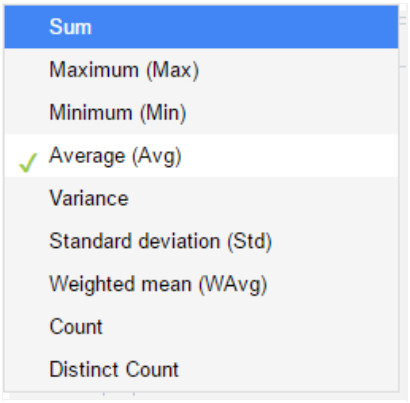
The fields configured in this totals can be edited in **Grid > Totals > Fields > Dynamic Group By**

Summary Totals

The fields added to the Summary totals will only be visible in Summary at the time Dynamic Group By is used.



All Grid fields can be added to the “totals”, the only difference being the functions available. Numeric fields can use all the available summarization functions, since the non-numeric fields can use only the count and the different count.



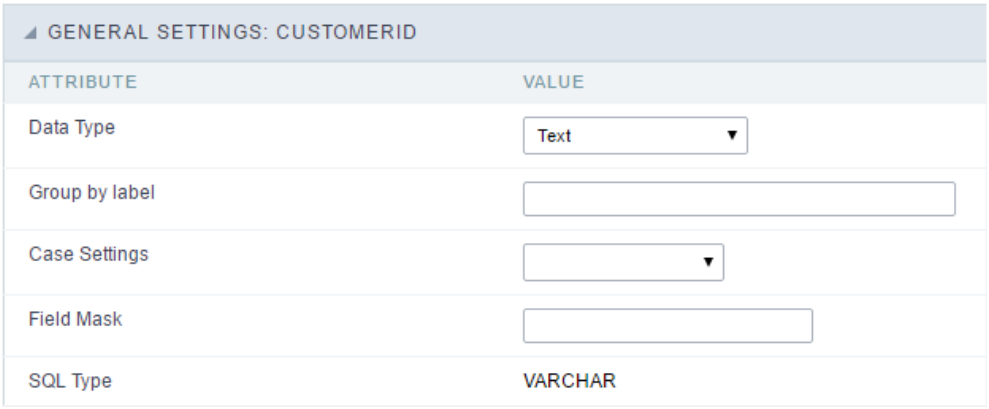
The fields configured in this totals can be edited in **Summary > Fields > Dynamic Group By**

Fields

The fields added to the **Grid Fields** will be listed within the Fields folder so that they can be edited individually.

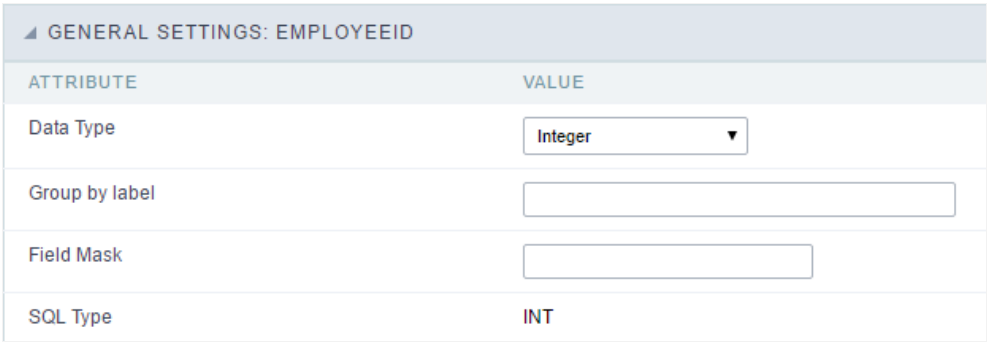
General Settings

Categories: Text and Barcode



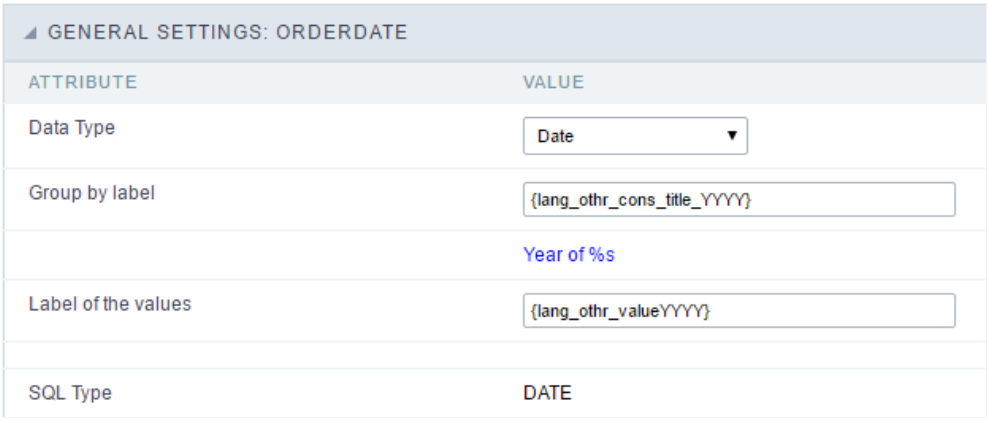
- **Data Type**: Defines the data type of the field in the HTML.
- **Group by label**: Defines a label of the fields displayed in the Group By.
- **Case Settings**: Converts the text according to the selected option.
  - **Upper Case**: Transforms the text in upper case.
  - **Lower Case**: Transforms the text in lower case.
  - **Capitalize first word**: It transforms the first word in the upper case or lower case, according to the text retrieved, that is, if the text retrieved by the application is in lower case, this option will make upper case only the first word.
  - **Capitalize all words**: Transforms all words in the upper case or lower case, according to the text retrieved, that is, if the text retrieved by the application is in lower case, this option will transform upper case into all words.
- **Field Mask**: On this field you will configure the display mask according to the table informed in Applications > Grid > Fields > Text.
- **SQL Type**: Informs the data type of the field in the database.

Categories: Number and Calculated



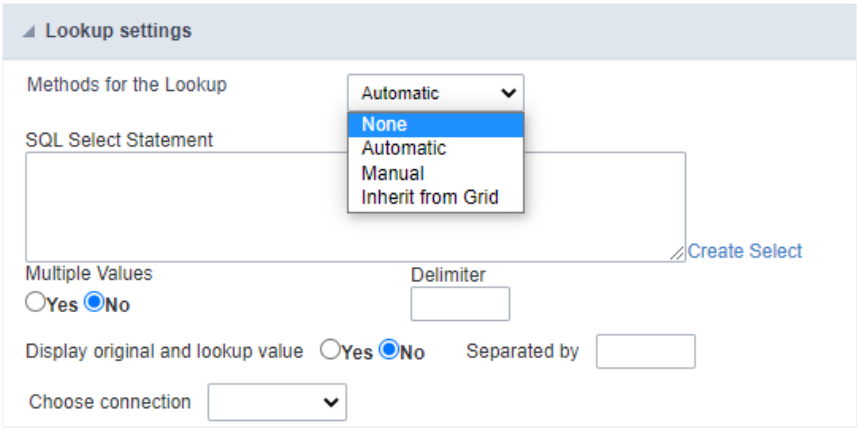
- **Data Type**: Defines the data type of the field in the HTML.
- **Group by label**: Defines a label of the fields displayed in the Group By.
- **Field Mask**: On this field you will configure the display mask according to the table informed in Applications > Grid > Fields > Text.
- **SQL Type**: Informs the data type of the field in the database.

Categories: Special and Date/Time



- **Data Type**: Defines the data type of the field in the HTML.
- **Group by label**: Defines a label of the fields displayed in the Group By.
- **SQL Type**: Informs the data type of the field in the database.

Lookup Settings



This option is available in the fields of the categories: Text, Number, Calculated, Special and Barcode.

On this option, it's possible to setup the display lookup for the fields cited above.

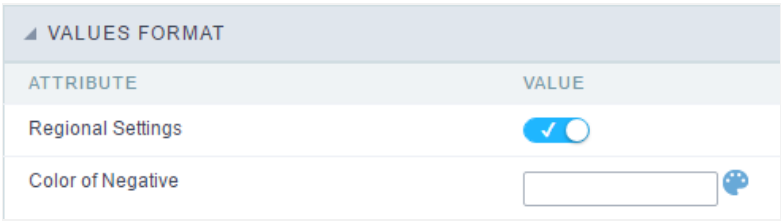
Three lookup methods are available.

- **Automatic** - SQL must be assembled to display the desired information. You can build a sql using the **Create select** option.
- **Manual** - The values that will be displayed in the application are informed in the interface. The value to be displayed (Label) and the corresponding value to the database (Value) must be informed.
- **Inherit from Grid** - Inherits the lookup setting configured in this field, in the field options of the grid application.

For more information about using the Grid Lookup, access Application > Grid > Fields > Text.

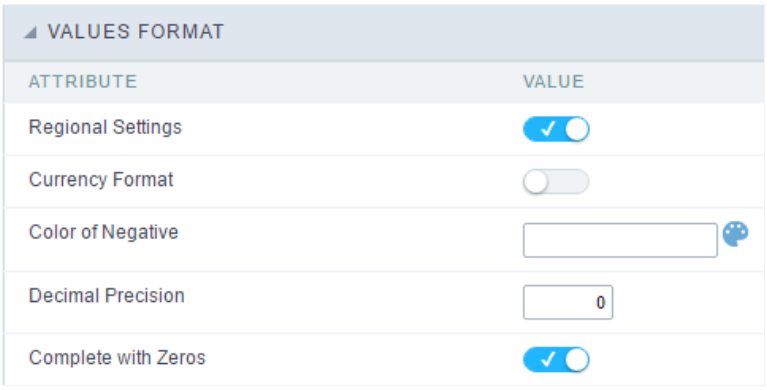
Values Format

Number Type



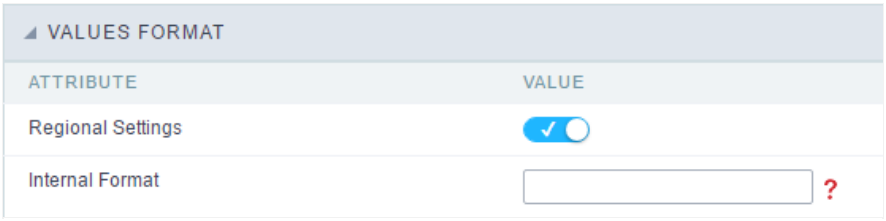
- **Regional Settings**: When active, you can apply the Regional Settings to this field. To modify the setting, access Locales > Regional Settings in the Scriptcase Menu.
- **Color of Negative**: In this field you'll inform the color value in hexadecimal. (Example: #000000)

Currency Type



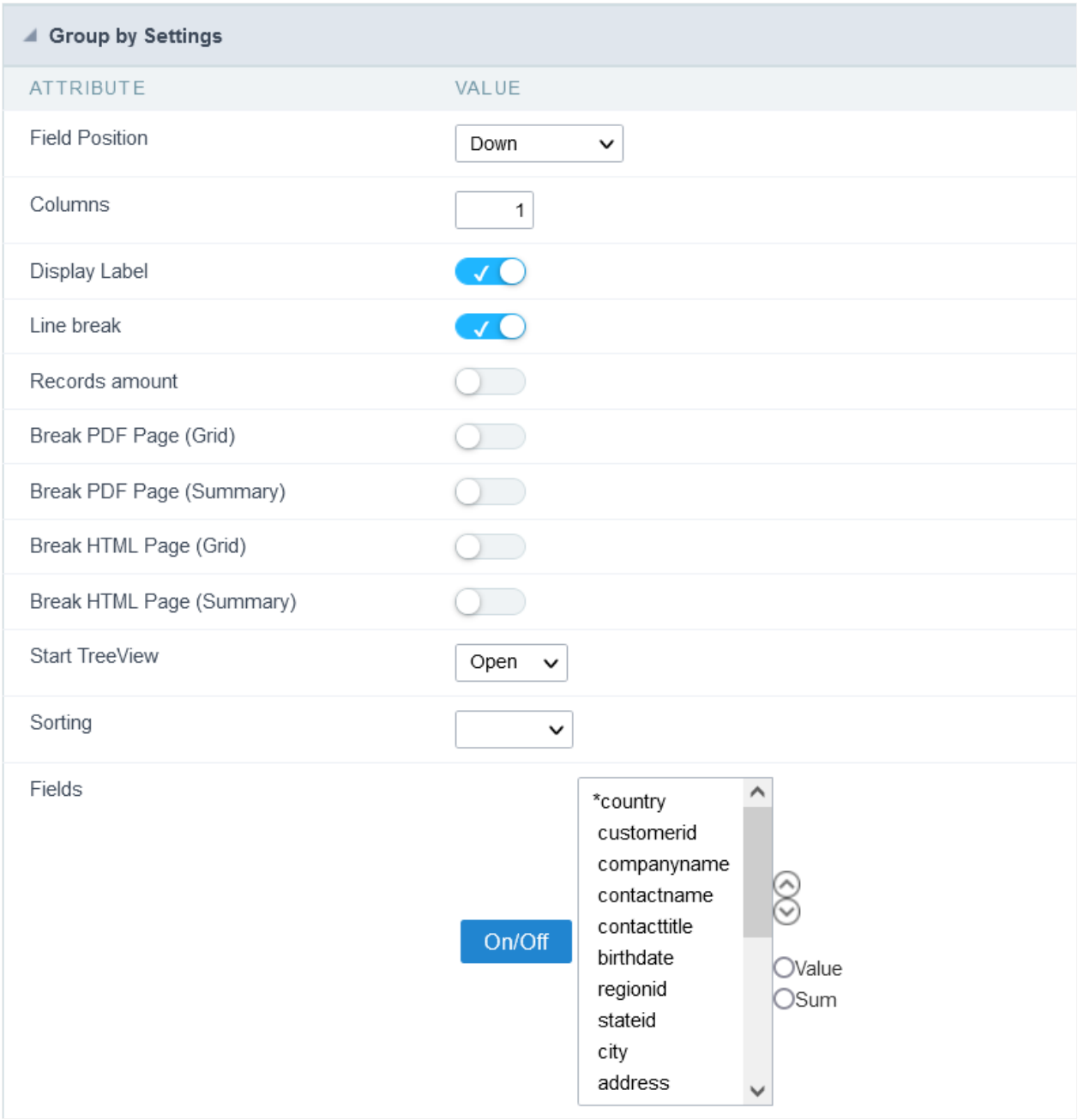
- **Regional Settings**: When active, you can apply the Regional Settings to this field. To modify the setting, access Locales > Regional Settings in the Scriptcase Menu.
- **Currency Symbol**: When this option is activated, the application will display the currency symbol according to the Regional Settings.
- **Color of Negative**: In this field you'll inform the color value in hexadecimal. (Example: #000000)
- **Decimal Precision**: Amount of decimal that your field will be displaying.
- **Complete with Zeros**: Activating this option will allow the application to complete the value after the coma with zeros.

Date Type



- **Regional Settings**: When active, you can apply the Regional Settings to this field. To modify the setting, access Locales > Regional Settings in the Scriptcase Menu.
- **Display**: In this field you will select how will the field will display itself.

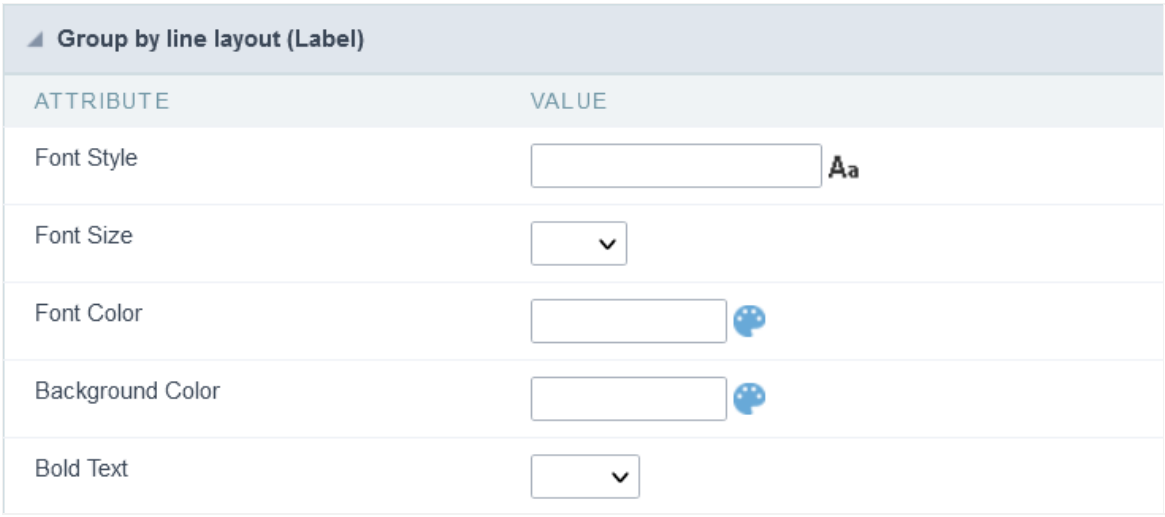
Group By Settings



- **Attributes**
  - **Field Position** : Defines how the information contained in the Group By line will be displayed and organized.
  - **Columns** : Defines the amount of columns information contained in the Group By line will displayed and organized.
  - **Display Label** : Defines if the field label will be displayed.

- **Line break** : Displays or not the Group By line with the value divided in another line.
- **Records amount** : Defines whether the Group By row will display the number of group records.
- **Break PDF Page (Grid)** : Defines if the PDF file, generated by the GRID, will contain each group by printed in a new page.
  - Ex. In a report you can have some orders where they can be displayed on different pages.
- **Break PDF Page (Summary)** : Defines if the Summary PDF file, generated by the GRID will contain each Group By printed in a new page.
  - Ex. In a report you can have some orders where they can be displayed on different pages. .
- **Break HTML Page (Grid)** : Displays each Grid Group By in a different HTML page.
- **Break HTML Page (Summary)** : Displays each Summary Group By in a different HTML page.
- **Start TreeView** : Defines the initial state of the TreeView.
- **Sorting** : If this option is set to “Yes”, the selected fields will sorted without the need to be clicked on, the sorting will be done respecting the criteria of the Group By.
- **Fields** : Defines which fields are going to displayed in the Group By row. It’s also possible to define if it’s going to be displayed in the Totals or in the Summary of this field by selecting Value or Sum, before clicking on the button “On/Off”.

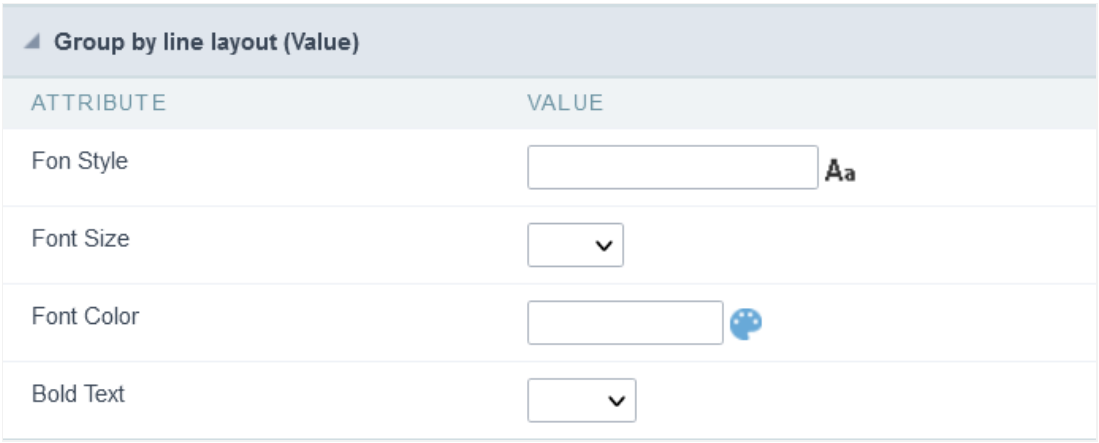
Group By Line Layout (Label)



Layout configuration of the Group By Label.

- **Attributes**
  - **Font Style** : Defines the font that’s going to be used in the Label.
  - **Font Size** : Defines the size of the font used in the Group By Label.
  - **Font Color** : Font color for the Group By Label.
  - **Background Color** : Group By background Color.
  - **Bold Text** : Defines if the label will have a Bold Text.

Group By Line Layout (Value)



Layout configuration of the Group By Value.

- **Attributes**
  - **Font Style** : Defines the font that is going to be used in the Value.
  - **Font Size** : Defines the size of the font used in the Group By Value.
  - **Font Color** : Font color for the Group By Value.
  - **Bold Text** : Defines if the value will have a Bold Text.

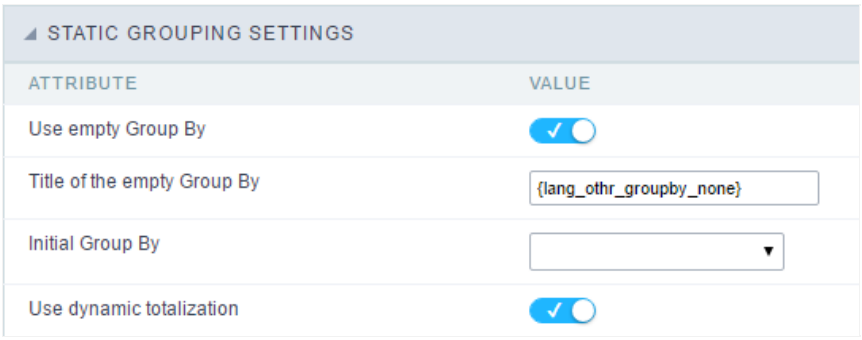
Static Group By

This type of Group By is configured by the developer, where the end user can select one of the predefined Group By available in the application.

Settings

These configurations affects the Static Group By.

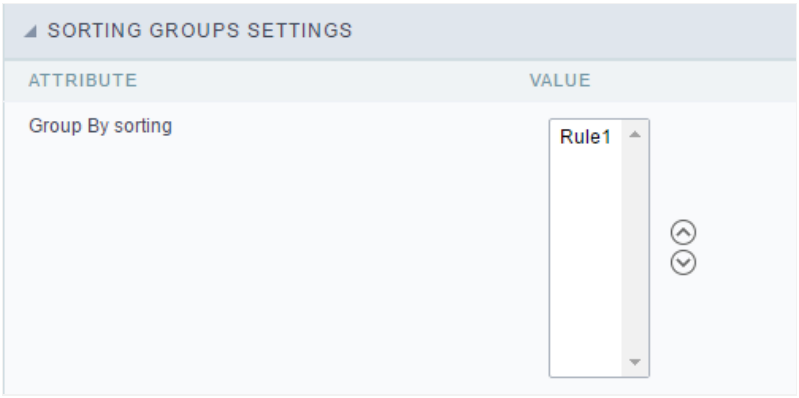
Static Group By Settings



- **Use empty Group By**: Configuration used when you want the Grid to be initialized without any Group By.
- **Title of the empty Group By**: Defines a title for the empty Group By option. (Available only when the previous option is enabled)
- **Initial Group By**: Defines the Group By used when running the generated application.

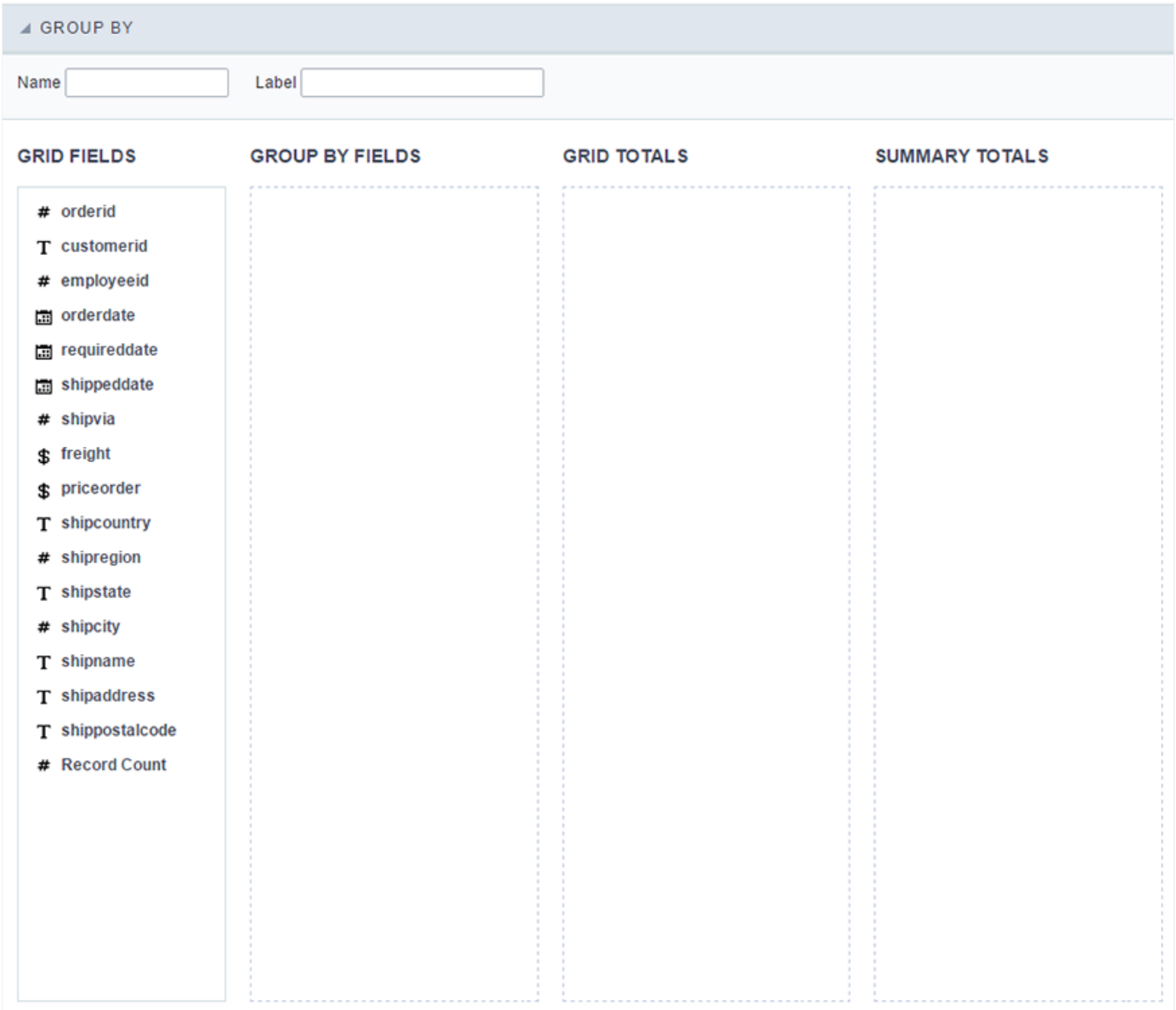
Sorting Groups Settings

Defines the order that the Group By's will be displayed in the Grid.



New Group By

Interface to create the static Group By.



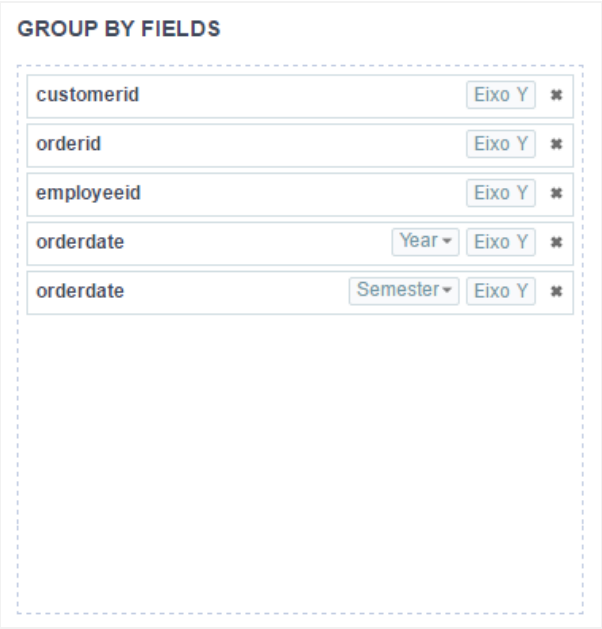
- **Name**: Defines the internal name of the Group By, used by ScriptCase.
- **Label**: Defines the displayed name of the Group By, that will be displayed in the application.

- **Grid Fields**: List of all the fields of the application.
- **Group By Fields**: Defines the fields that are part of the Group By.

- **Grid Totals**: Defines the fields that are part of the Grid Totals.
- **Summary Totals**: Defines the fields that are part of the Summary Totals.

Group By Fields

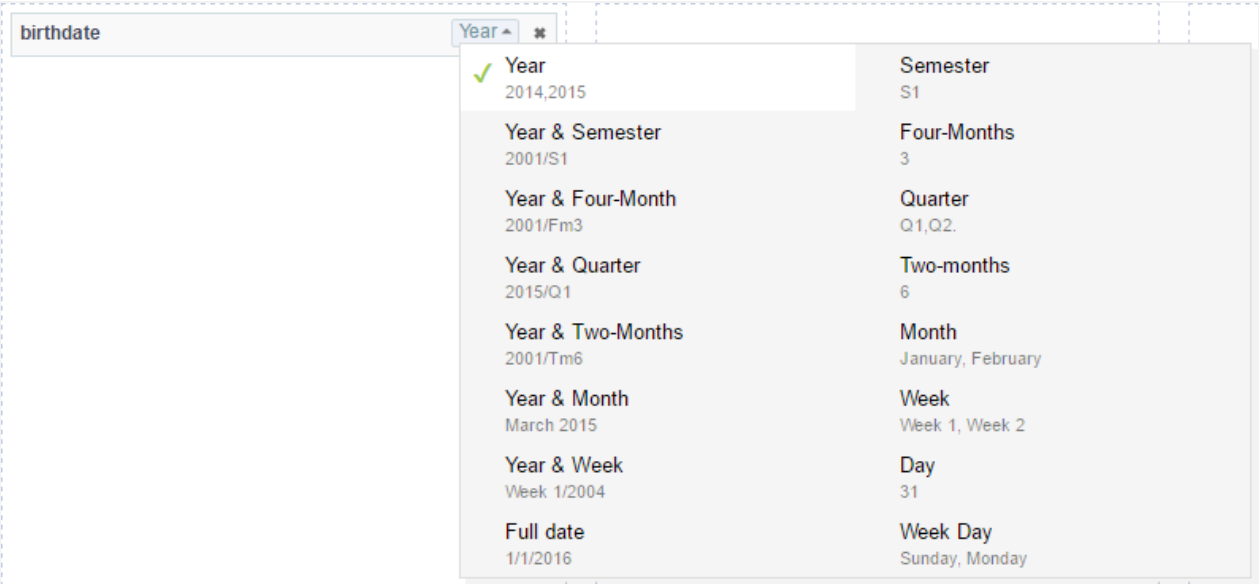
You need to drag and drop the fields that'll be part of the Group By in the **Group By Fields**.



Each field can only be added once to the **Group By Fields**, except for the date and datetime fields.

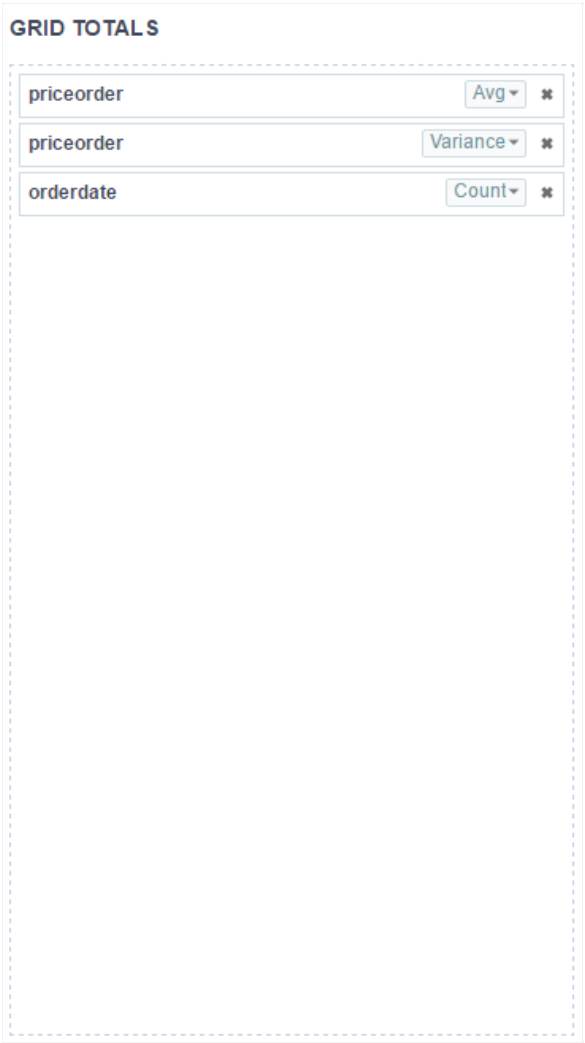
Date and Datetime fields

For date and datetime fields, some display intervals have been added, so these fields can be added two or more times to the Group By.

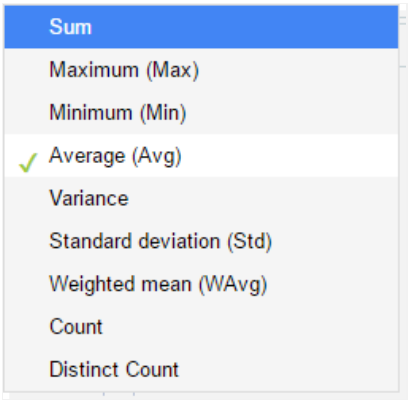


Grid Totals

The fields added in this total will be visible only in the Grid (if the total field is visible in the Grid) at the time that the group by is used.



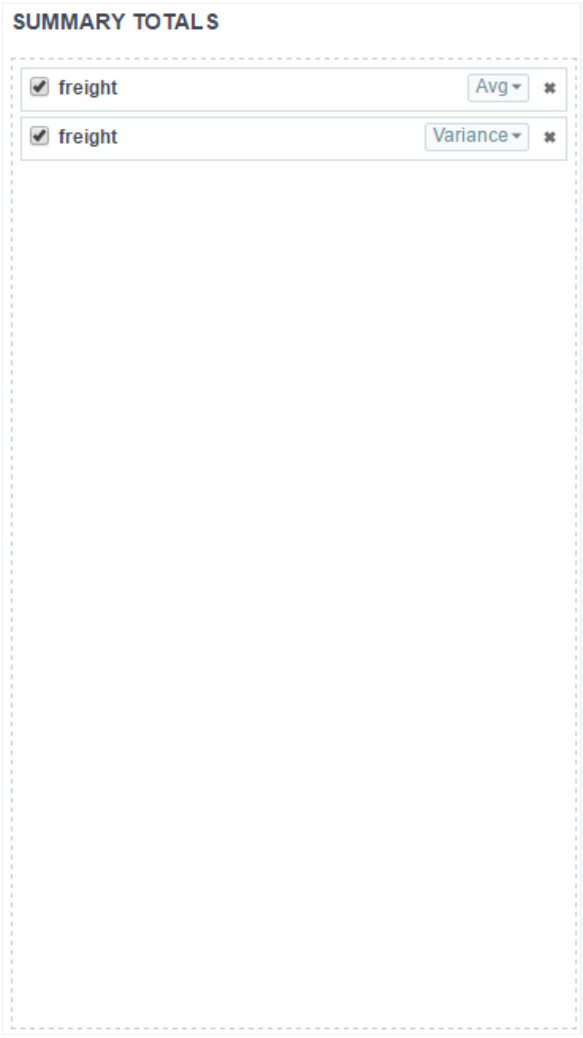
All Grid fields can be added to the “totals”, the only difference being the functions available. Numeric fields can use all the available summarization functions, since the non-numeric fields can use only the count and the different count.



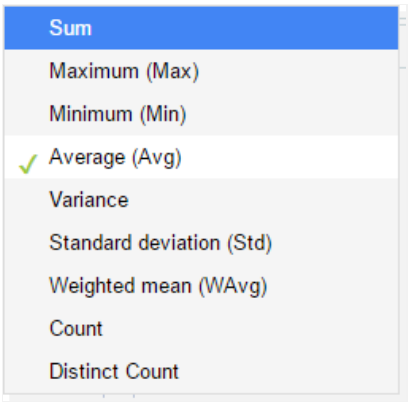
The fields configured in this totals can be edited in **Grid > Totals > Fields > Dynamic Group By**

Summary Totals

The fields added to the Summary totals will only be visible in Summary at the time Dynamic Group By is used.



All Grid fields can be added to the “totals”, the only difference being the functions available. Numeric fields can use all the available summarization functions, since the non-numeric fields can use only the count and the different count.



The fields configured in this totals can be edited in **Summary > Fields > Dynamic Group By**

Edit Group By

After the creation steps, all the Group By will be listed in the **Static Group By** folder, below the Settings item.

Expanding the Group By Folder, you will see the **settings** icon, where you can edit the entire Group By.

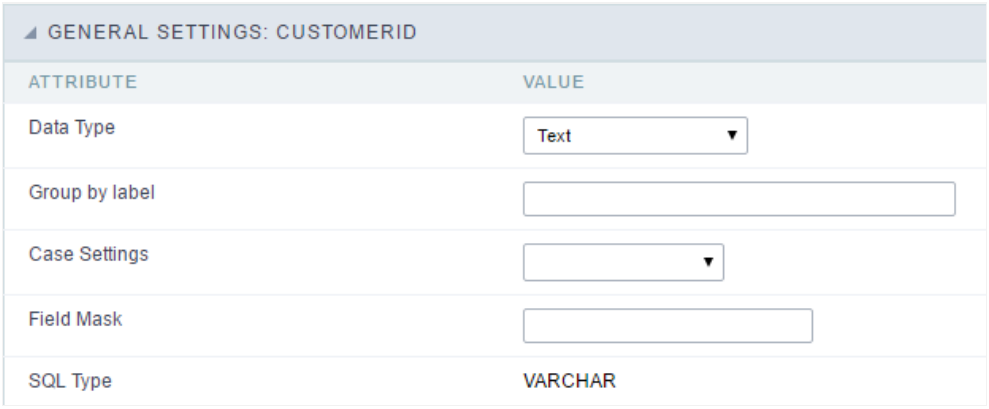
You can also view the fields used in the Group By, that can be edited independently from the Grid Layout, see the following:

Fields

The fields added to the **Grid Fields** will be listed within the Fields folder so that they can be edited individually.

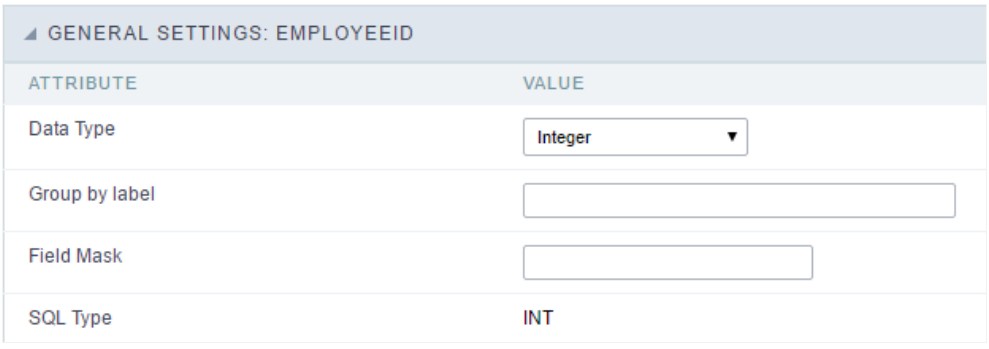
General Settings

Categories: Text and Barcode



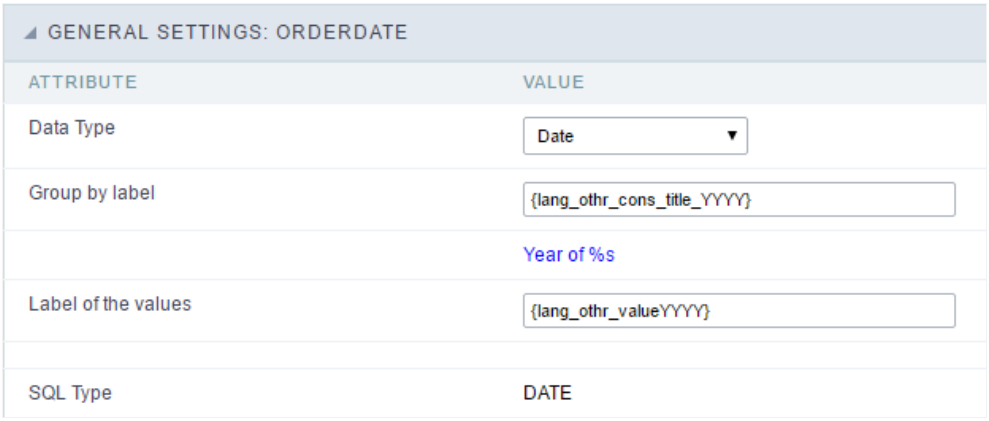
- **Data Type**: Defines the data type of the field in the HTML.
- **Group by label**: Defines a label of the fields displayed in the Group By.
- **Case Settings**: Converts the text according to the selected option.
  - **Upper Case**: Transforms the text in upper case.
  - **Lower Case**: Transforms the text in lower case.
  - **Capitalize first word**: It transforms the first word in the upper case or lower case, according to the text retrieved, that is, if the text retrieved by the application is in lower case, this option will make upper case only the first word.
  - **Capitalize all words**: Transforms all words in the upper case or lower case, according to the text retrieved, that is, if the text retrieved by the application is in lower case, this option will transform upper case into all words.
- **Field Mask**: On this field you will configure the display mask according to the table informed in Applications > Grid > Fields > Text.
- **SQL Type**: Informs the data type of the field in the database.

Categories: Number and Calculated



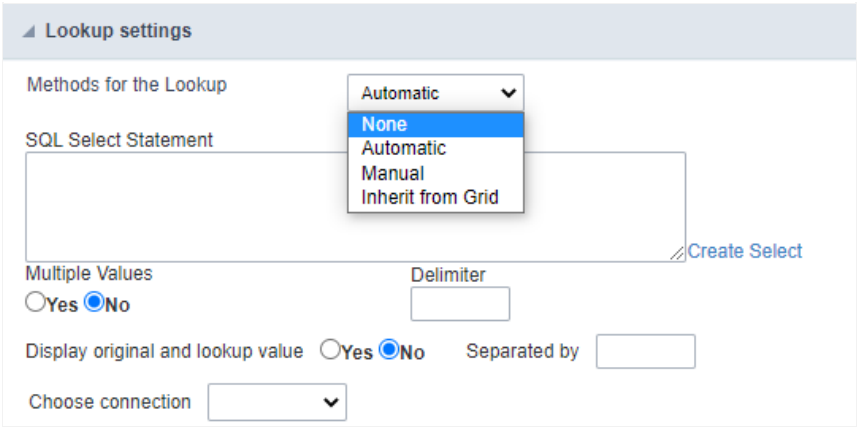
- **Data Type**: Defines the data type of the field in the HTML.
- **Group by label**: Defines a label of the fields displayed in the Group By.
- **Field Mask**: On this field you will configure the display mask according to the table informed in Applications > Grid > Fields > Text.
- **SQL Type**: Informs the data type of the field in the database.

Categories: Special and Date/Time



- **Data Type**: Defines the data type of the field in the HTML.
- **Group by label**: Defines a label of the fields displayed in the Group By.
- **SQL Type**: Informs the data type of the field in the database.

Lookup Settings



This option is available in the fields of the categories: Text, Number, Calculated, Special and Barcode.

On this option, it's possible to setup the display lookup for the fields cited above.

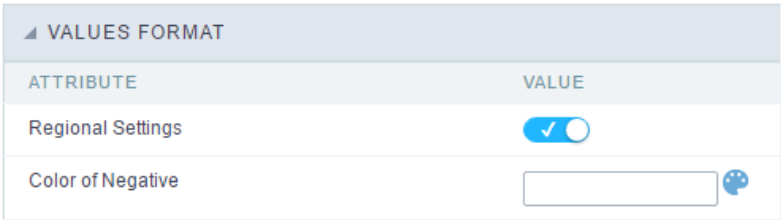
Three lookup methods are available.

- **Automatic** - SQL must be assembled to display the desired information. You can build a sql using the **Create select** option.
- **Manual** - The values that will be displayed in the application are informed in the interface. The value to be displayed (Label) and the corresponding value to the database (Value) must be informed.
- **Inherit from Grid** - Inherits the lookup setting configured in this field, in the field options of the grid application.

For more information about using the Grid Lookup, access Application > Grid > Fields > Text.

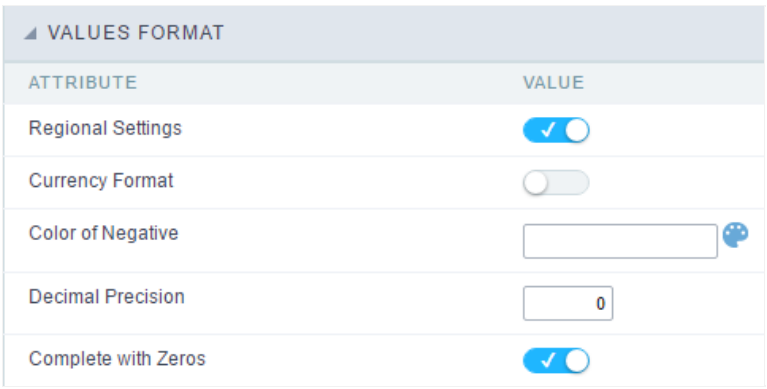
Values Format

Number Type



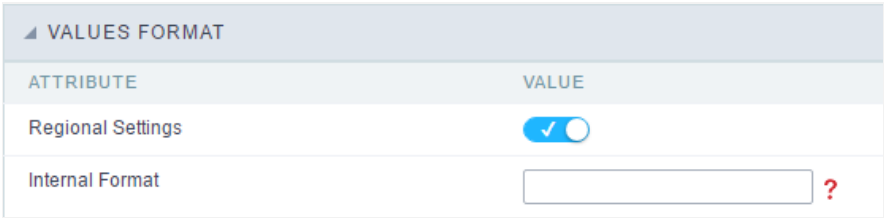
- **Regional Settings**: When active, you can apply the Regional Settings to this field. To modify the setting, access Locales > Regional Settings in the Scriptcase Menu.
- **Color of Negative**: In this field you'll inform the color value in hexadecimal. (Example: #000000)

Currency Type



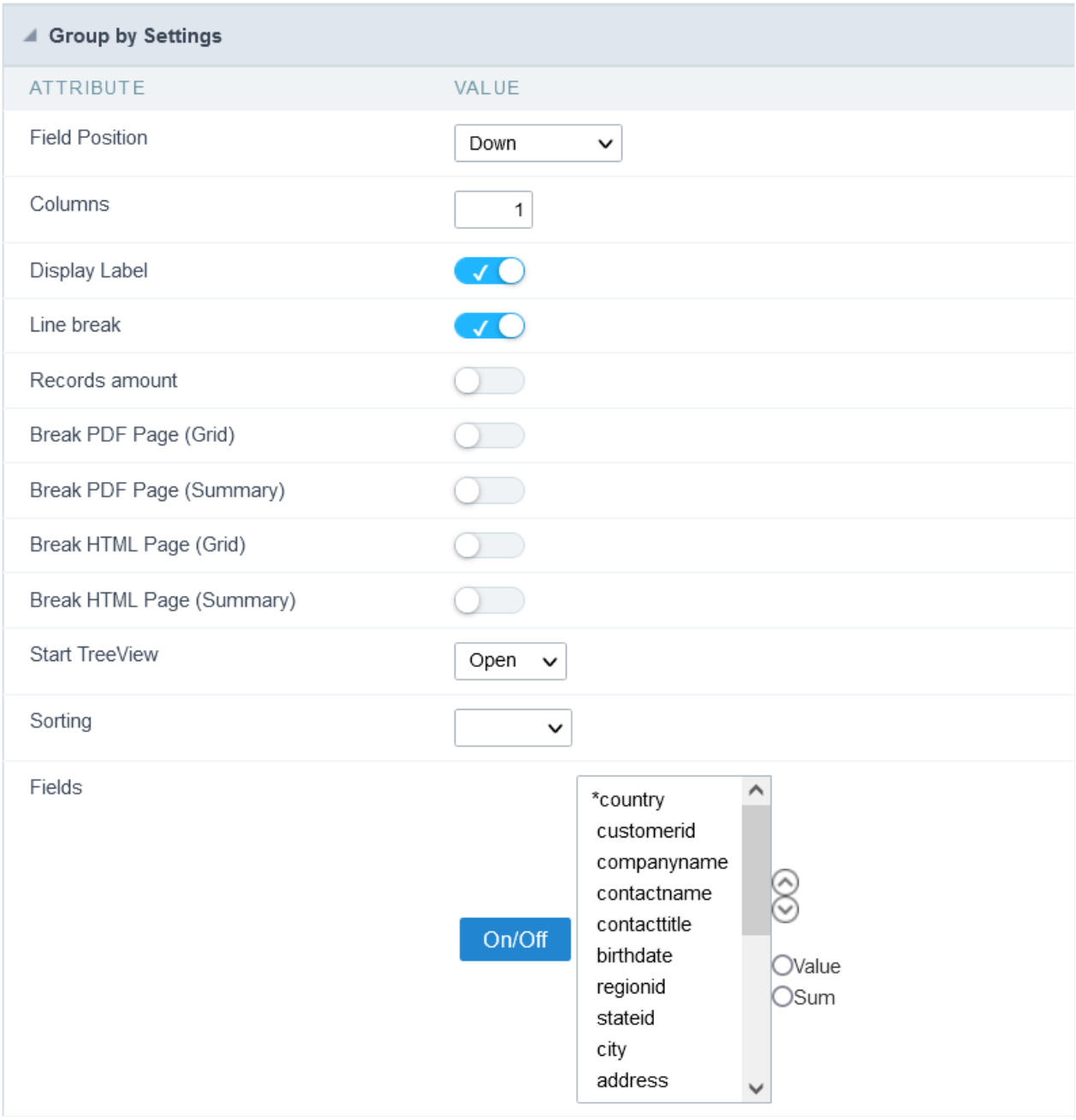
- **Regional Settings**: When active, you can apply the Regional Settings to this field. To modify the setting, access Locales > Regional Settings in the Scriptcase Menu.
- **Currency Symbol**: When this option is activated, the application will display the currency symbol according to the Regional Settings.
- **Color of Negative**: In this field you'll inform the color value in hexadecimal. (Example: #000000)
- **Decimal Precision**: Amount of decimal that your field will be displaying.
- **Complete with Zeros**: Activating this option will allow the application to complete the value after the coma with zeros.

Date Type



- **Regional Settings**: When active, you can apply the Regional Settings to this field. To modify the setting, access Locales > Regional Settings in the Scriptcase Menu.
- **Display**: In this field you will select how will the field will display itself.

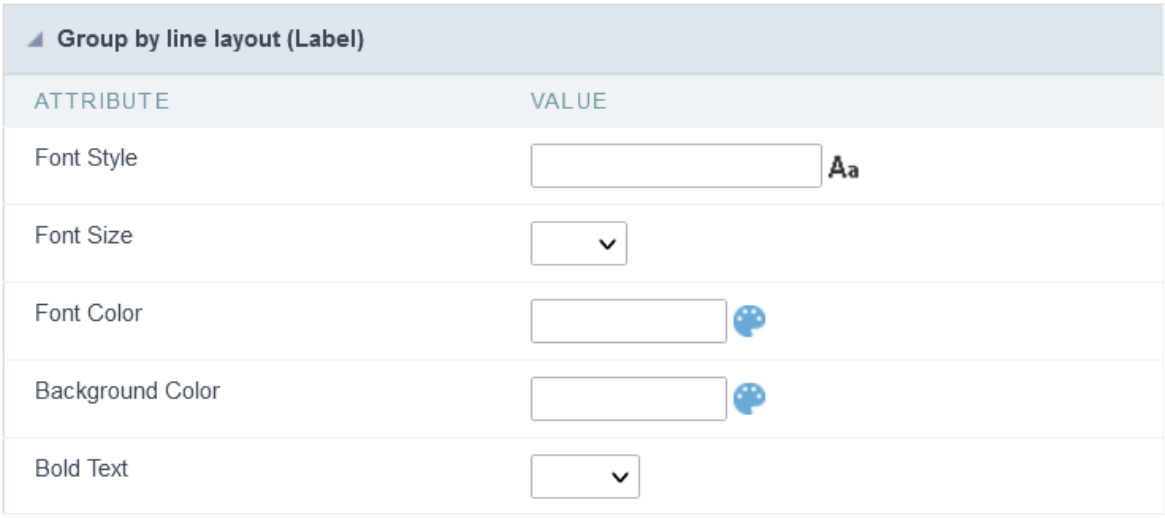
Group By Settings



- **Attributes**
  - **Field Position** : Defines how the information contained in the Group By line will be displayed and organized.
  - **Columns** : Defines the amount of columns information contained in the Group By line will displayed and organized.
  - **Display Label** : Defines if the field label will be displayed.
  - **Line break** : Displays or not the Group By line with the value divided in another line.
  - **Records amount** : Defines whether the Group By row will display the number of group records.
  - **Break PDF Page (Grid)** : Defines if the PDF file, generated by the GRID, will contain each group by printed in a new page.
    - Ex. In a report you can have some orders where they can be displayed on different pages.
  - **Break PDF Page (Summary)** : Defines if the Summary PDF file, generated by the GRID will contain each Group By printed in a new page.
    - Ex. In a report you can have some orders where they can be displayed on different pages. .
  - **Break HTML Page (Grid)** : Displays each Grid Group By in a different HTML page.
  - **Break HTML Page (Summary)** : Displays each Summary Group By in a different HTML page.
  - **Start TreeView** : Defines the initial state of the TreeView.
  - **Sorting** : If this option is set to “Yes”, the selected fields will sorted without the need to be clicked on, the sorting will be done respecting the criteria of the Group By.
  - **Fields** : Defines which fields are going to displayed in the Group By row. It’s also possible to define if it’s going to be

displayed in the Totals or in the Summary of this field by selecting Value or Sum, before clicking on the button “On/Off”.

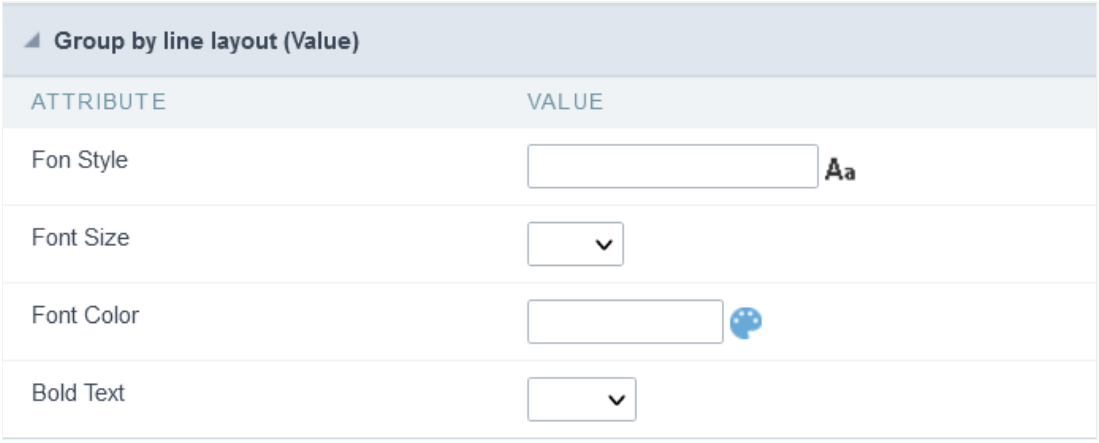
Group By Line Layout (Label)



Layout configuration of the Group By Label.

- **Attributes**
  - **Font Style** : Defines the font that’s going to be used in the Label.
  - **Font Size** : Defines the size of the font used in the Group By Label.
  - **Font Color** : Font color for the Group By Label.
  - **Background Color** : Group By background Color.
  - **Bold Text** : Defines if the label will have a Bold Text.

Group By Line Layout (Value)



Layout configuration of the Group By Value.

- **Attributes**
  - **Font Style** : Defines the font that is going to be used in the Value.
  - **Font Size** : Defines the size of the font used in the Group By Value.
  - **Font Color** : Font color for the Group By Value.
  - **Bold Text** : Defines if the value will have a Bold Text.

Events

These events are only available for static Group By.

The **OnGroupByAll** event occurs after running the Group By, that allows you to manipulate the totals variables.

Total variables are created based on the fields selected for group by and totals.

The following is an example of the available variables:

Assuming that an application has a Group By **state** and **city** and two summarization per **parcel** and **balance**, the following summarization variables are available:

Variables	Description
{count_ger}	Contains the total number of records.
{sum_parcel}	Contains the general sum of the parcel field.
{sum_balance}	Contains the overall sum of the balance field.
{count_state}	Contains the total amount of records, from the state group by being processed.
{sum_parcel_state}	Contains the general sum of the 'state' field and the 'parcel' field that are being processed.
{sum_balance_state}	Contains the general sum of the 'balance' field and the 'parcel' field that are being processed.
{count_city}	Contains the total number of records, of the city group by being processed.
{sum_parcel_city}	Contains the general sum of the 'parcel' and 'city' field that are being processed.
{sum_balance_city}	Contains the general sum of the 'balance' and 'city' fields that are being processed.

To access the summarization variables by group, simply replace the Group By name with **Group By**. Example: **{sum_balance_city}** for **{sum_quebra_balance}**

Variables	Description
{count_quebra}	The total number of records in the Group By that is being processed.
{sum_quebra_parcel}	Contains the general sum of the parcel field, of the Group By that is being processed.
{sum_quebra_balance}	Contains the sum total of the balance field, of Group By that is being processed.

Example:

In an application that has Group By by state and city and totals a balance field in Group By totals, we want to display the average in place of the balance. A method is created in the **OnGroupByAll** event, with the following content:

```
{sum_quebra_balance} = {sum_quebra_balance} / {count_quebra};
```

Related Links 🔗

Related Video ▷

Totals

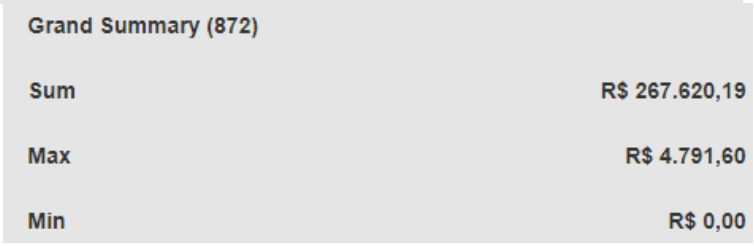
The total fields will be displayed only when the Grid application is using at least one Group By.

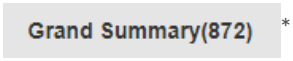
Settings

The settings below can be applied for the Grid totals only.

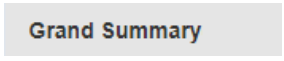
- **Results in a single line.**: This option sets the display of the **General Total** title and its results in a single line.
  - Example for this option **Enabled**:

  - Example for this option **Disabled**:



- **Display Total** : This option indicates in which pages the General Totals will be displayed. The options are: **On every page**, **On the last page** or **Do not display**. *__Group Subtotal__ : This option defines where the subtotal of group by will be displayed.
- **Record Count** : This option allows you to view the amount of records by the General Total title.
  - Example for this option **Enabled**: 

* Example for this option **Disabled**:



Layout settings

Display settings of the group subtotal.

- **Label** - This field sets the Label for the **Group Subtotal**.
- **Display** - This option sets if the **Group Subtotal** will be displayed.

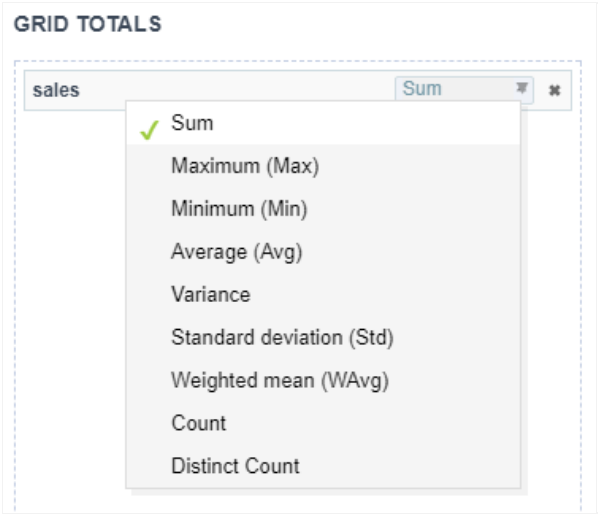
Select Fields

Using the select fields area you can set the total fields ant the total options for each field. The same field can be used more than once in the Grid totals area, just if they are using different summaries types.

To define the fields that will be used for totals, drag them to the area, **Grid Totals**.

Note: The total fields will only be displayed if they are also selected to be displayed within the Grid application module.

When positioning the fields, you must define what summarization will be used, to select that you must click in the combo box and select one of the available options, that will be according to the data type (integer, date, text...).



The summarization options available are:

- **Sum** : Sets a sum of the values for the selected field.
- **Maximum** : Displays the highest value identified in the selected field.
- **Minimum** : Displays the lowest value identified in the selected field.
- **Average (Avg)** : Calculates the arithmetic mean of the values for the selected field.

- **Variance** : Calculates the dispersion of the values related to the average.
- **Standard Deviation** : Measures the variability of values around the average, the minimum value of the standard deviation is 0 indicating that there is variability, i.e. that all values are equal to the mean.
- **Weighted mean (WAvg)** : Calculates the weighted average for the selected field. To set the weight used in the calculation of the average access field settings selected in **Totals > Fields (select the field where you are using the Weighted mean) > Weighted average weight**.
    - **Weighted average weight** : Field that will be used as weighted average weight. In calculating the weighted average, each set value is multiplied by its “weight”, that is its relative importance.
- **Count** : Displays the total number of records for the selected field.
- **Distinct Count** : Displays the total number of records for the selected field, distinguishing the values.

Important note: The fields in the Grid totals are displayed only on Grids with no Group by or with empty Group by.

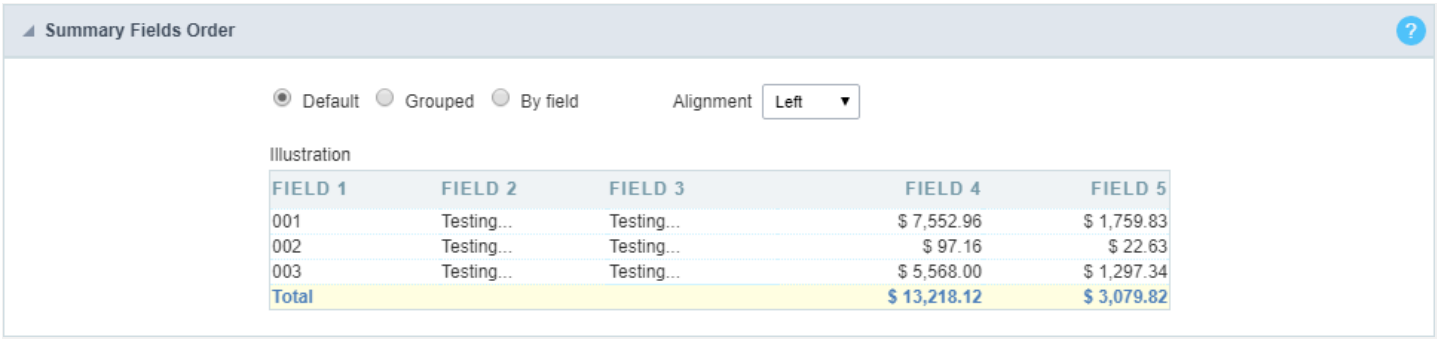
Positioning

Summary Fields Order - Default

Defines the positioning and the label used by Grid totals. There are three display formats,, **Default**, **Grouped** or **By field**

To add the same field two or more times in the Totals or use different types of summaries in selected fields, this option will not be displayed.

The option **Default** returns the result below the column being summarized. When you use the total for more than one column of the Grid, using the same type of summarization, the results are displayed on the same line.



The line with **General Total** displaying the **Sales**.

Id	Date	Region	Product Category	Sales
1	01/01/2015	South Atlantic	SEAFOOD	R$ 50,82
2	01/01/2016	South Atlantic	CONFECTIONS	R$ 84,70
3	01/01/2013	South Atlantic	BEVERAGES	R$ 90,75
4	01/02/2013	East South Central	CONFECTIONS	R$ 163,35
5	01/02/2014	East South Central	DAIRY PRODUCTS	R$ 251,68
Grand Summary				**R$ 267.620,19**

Alignment

This option is available for display formats **Default** and **Grouped**.

Defines the placement of the label selected within the **Setting > Display total** option. The alignment can be in the **Center**, **Left** e **Right**:

- **Left** :

Grand Summary	R$ 267.620,19

- **Right** :

Grand Summary	R$ 267.620,19

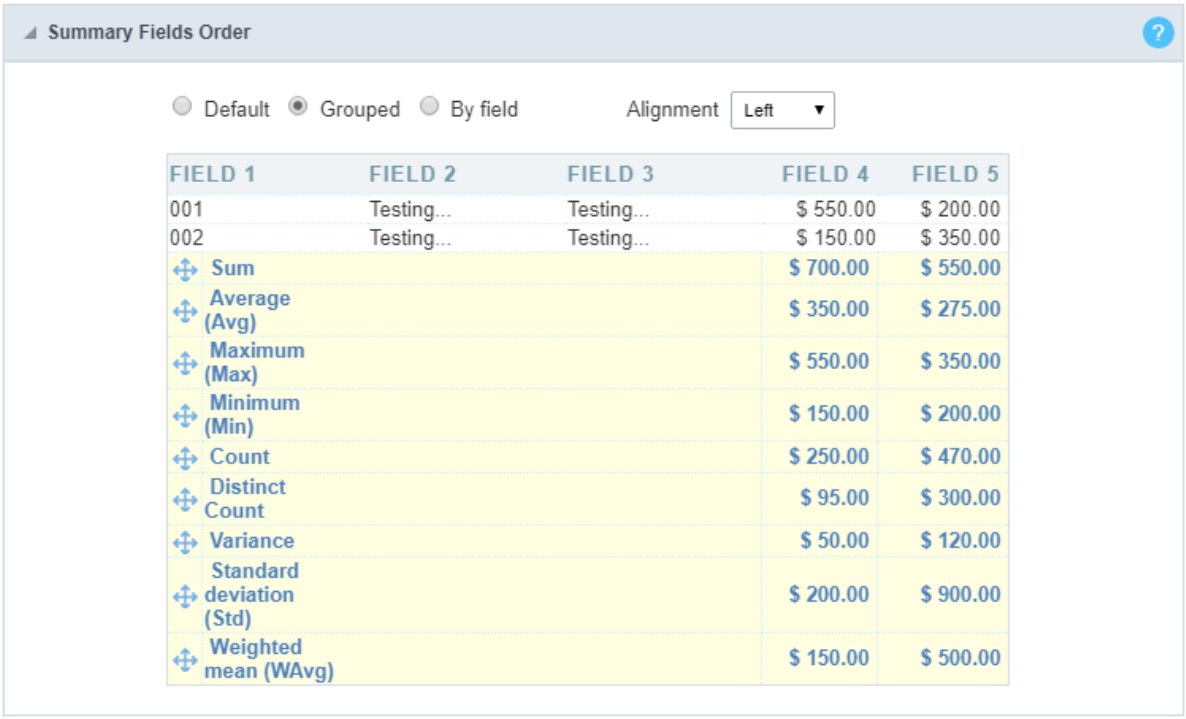
- **Center** :

Grand Summary	R$ 267.620,19

Positioning - Grouped

The option **Grouped** returns the total result, separating each type per line.

It is possible to position the total lines by dragging to the desired position.



FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5
001	Testing...	Testing...	$ 550.00	$ 200.00
002	Testing...	Testing...	$ 150.00	$ 350.00
Sum			$ 700.00	$ 550.00
Average (Avg)			$ 350.00	$ 275.00
Maximum (Max)			$ 550.00	$ 350.00
Minimum (Min)			$ 150.00	$ 200.00
Count			$ 250.00	$ 470.00
Distinct Count			$ 95.00	$ 300.00
Variance			$ 50.00	$ 120.00
Standard deviation (Std)			$ 200.00	$ 900.00
Weighted mean (WAvg)			$ 150.00	$ 500.00

In this example, we are displaying the sum, average, maximum and minimum of column **Sales**.

Id	Date	Region	Product Category	Sales
1	01/01/2015	South Atlantic	SEAFOOD	R$ 50,82
2	01/01/2016	South Atlantic	CONFECTIONS	R$ 84,70
3	01/01/2013	South Atlantic	BEVERAGES	R$ 90,75
4	01/02/2013	East South Central	CONFECTIONS	R$ 163,35
5	01/02/2014	East South Central	DAIRY PRODUCTS	R$ 251,68
Grand Summary				
Sum				**R$ 267.620,19**
Avg				**R$ 306,90**
Max				**R$ 4.791,60**
Min				**R$ 0,00**

Alignment

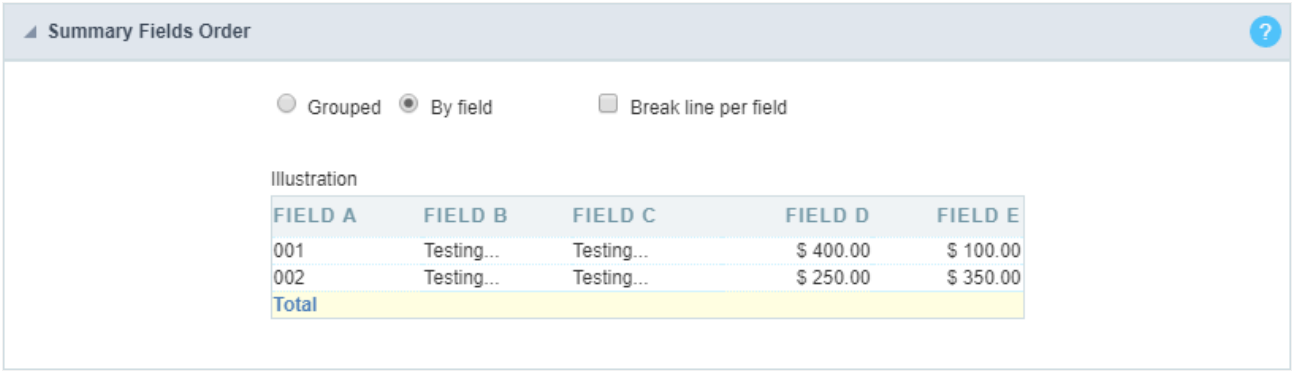
This option is available for display formats **Default** and **Grouped**.

Defines the placement of the label selected within the **Setting > Display total** option. The alignment can be in the **Center**, **Left** e **Right**:

- **Left** : **Grand Summary** **R$ 267.620,19**
- **Right** : **Grand Summary** **R$ 267.620,19**
- **Center** : **Grand Summary** **R$ 267.620,19**

Positioning - By field

This option displays the values in the left corner by positioning the results next to each other.

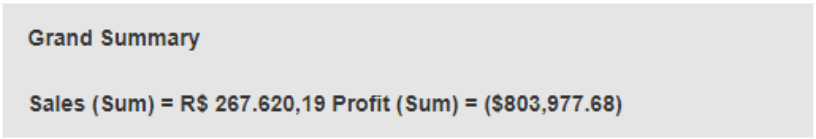


Line break per field

This option sets the total fields display. If selected, it will show the fields in the same row or divided by line.

Separated by line:

Displayed on the same line:

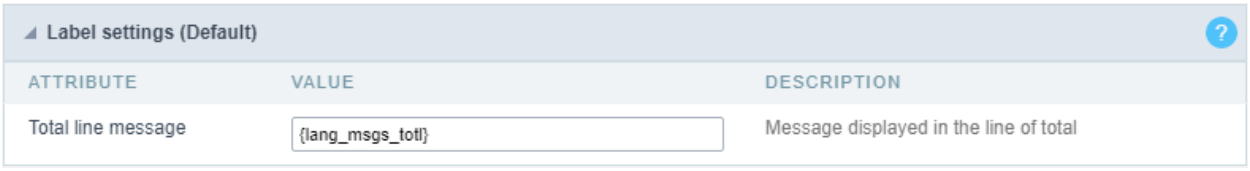


Label Settings

Label settings (Default)

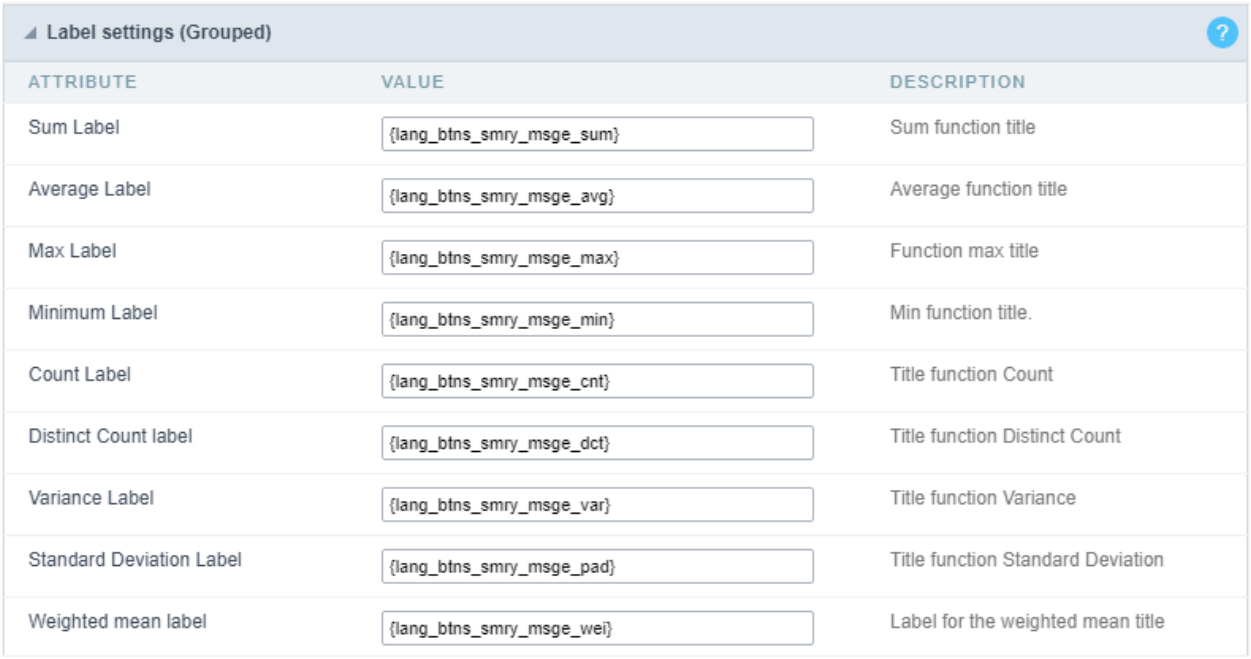
This option allows you to customize the total labels.

As default we are going to display Grant totals



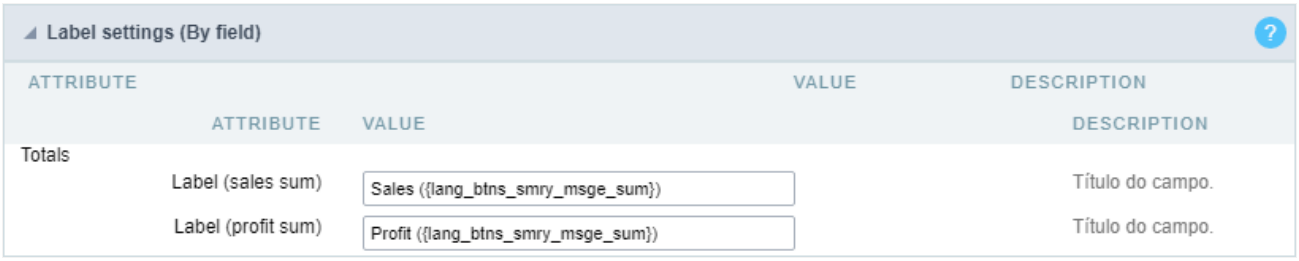
Label settings (Grouped)

This option allows you to customize the total labels.



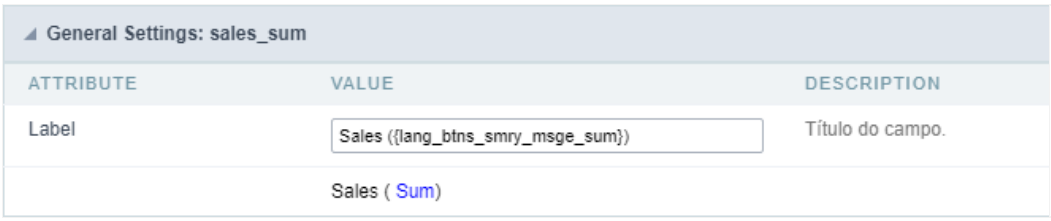
Label settings (By field)

This option allows you to customize the total labels.



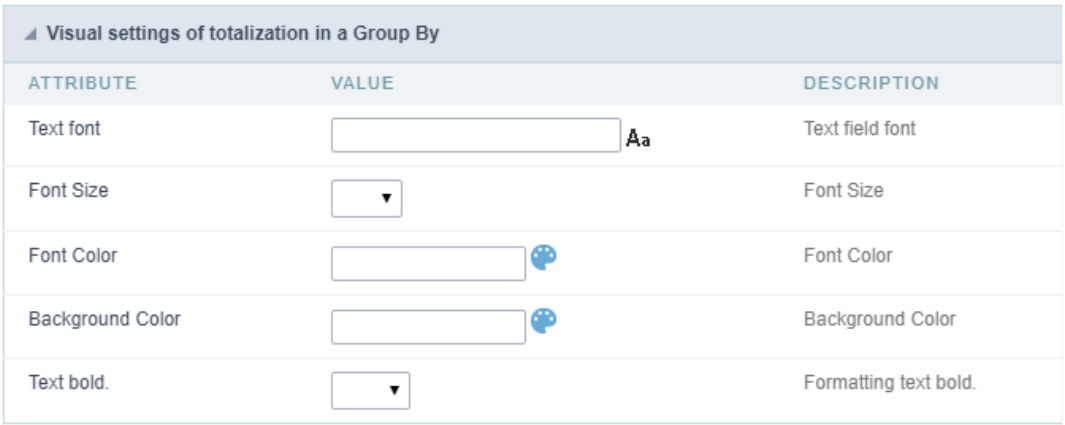
Fields - General Settings

Allows you to change the label displayed in the totals.



Fields - Visual Settings in a Group by

Sets the formatting of the fields displayed in the subtotal for the group.



- **Font family** : Sets the font used.
- **Font size** : Sets the font size.
- **Text color** : Sets the text color
- **Background color** : Sets the background color.
- **Bold** : Format text in bold.

Example of formatting the subtotal for the Group:

Date of Date => 01/09/2013

Id	Date	Region	Product Category	Sales	Profit
22	01/09/2013	East South Central	BEVERAGES	R$ 163,35	$36.30
19	01/09/2013	East South Central	GRAINS/CEREALS	R$ 363,00	$121.00
25	01/09/2013	East North Central	CONFECTIONS	R$ 116,16	$19.36
Total sales on the day - 01/09/2013				R$ 642,51	

Fields - General Totals Visual Setting

Sets the formatting of the fields displayed in the General total.

Visual setting of the General Total

ATTRIBUTE	VALUE	DESCRIPTION
Text Font	Aa	Field Text size
Font Size		Font Size
Font Color.		Font Color.
Background Color		Background Color
Bold text		Formating text in bold.

- **Font family** : Sets the font used.
- **Font size** : Sets the font size.
- **Text color** : Sets the text color
- **Background color** : Sets the background color.
- **Bold** : Format text in bold.

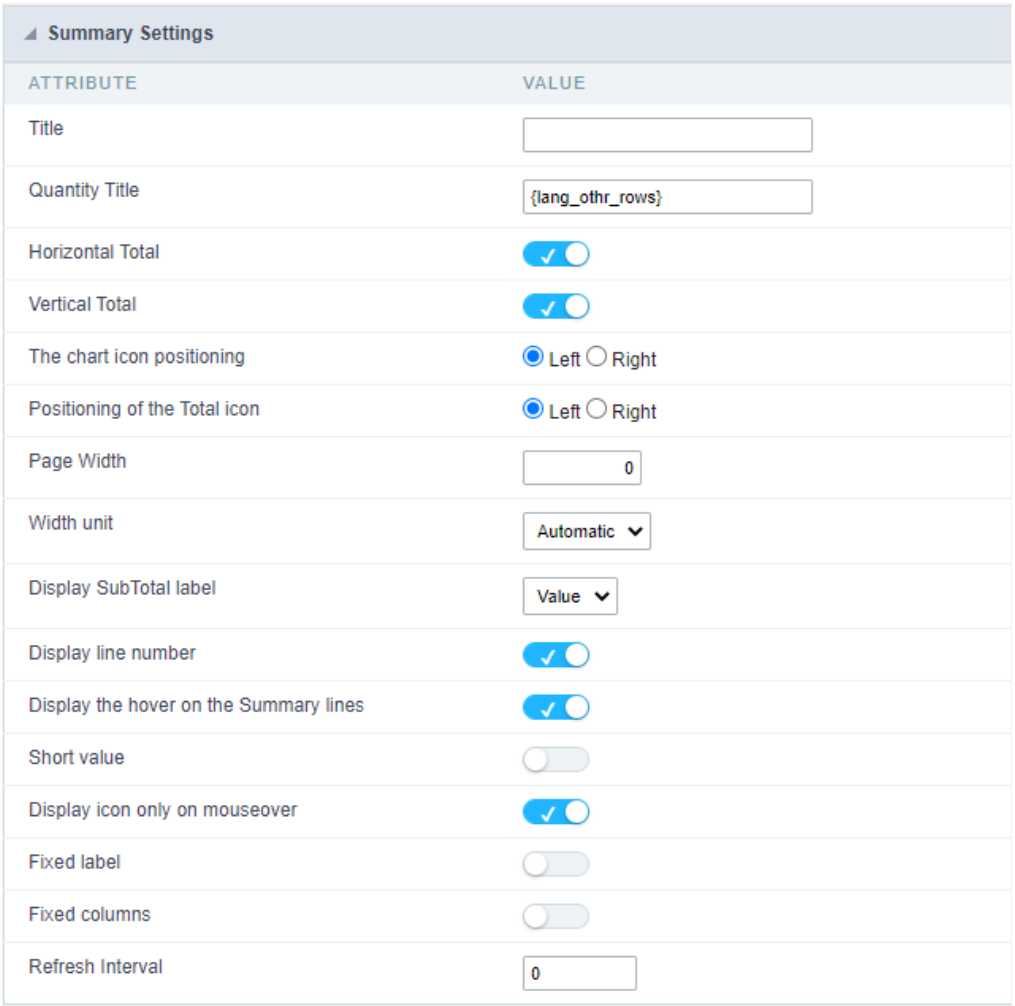
Example of formatting the Grid General Total

Id	Date	Region	Product Category	Sales	Profit
1	01/01/2015	South Atlantic	SEAFOOD	R$ 50,82	$14.52
2	01/01/2016	South Atlantic	CONFECTIONS	R$ 84,70	$33.88
3	01/01/2013	South Atlantic	BEVERAGES	R$ 90,75	$36.30
4	01/02/2013	East South Central	CONFECTIONS	R$ 163,35	$54.45
5	01/02/2014	East South Central	DAIRY PRODUCTS	R$ 251,68	$94.38
Grand Summary				R$ 267.620,19	

Related Links

Summary

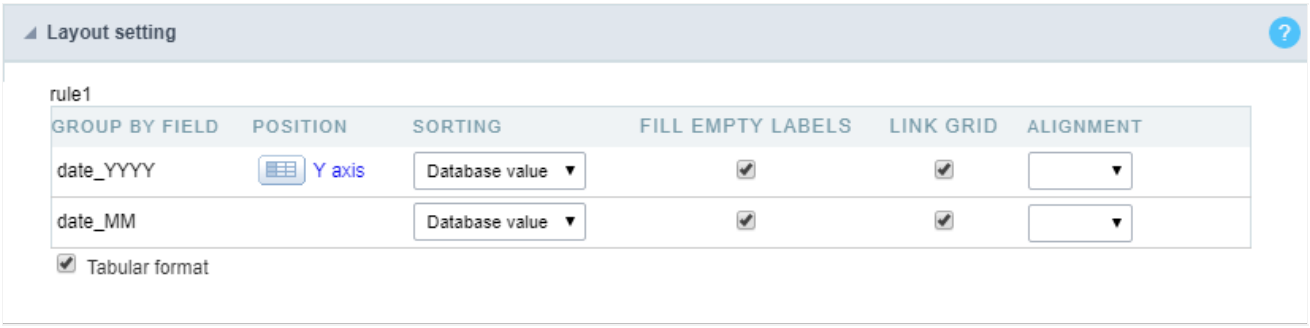
Summary Settings



- **Title**: Defines a title for the Summary, if not informed the default title will be “Summary”. For no title to be displayed, enter the HTML tag ` ` .
- **Quantity Title**: Title Amount of Record.
- **Horizontal Total**: Show horizontal total for Summaries of matrix type.
- **Vertical Total**: Display vertical total for Summaries of matrix type.
- **The chart icon positioning**: Chart icon positioning (left or right).
- **Positioning of the Total icon**: Total icon positioning (left or right).
- **Page Width**: Width value for the Summary page.
- **Width unit**: Unit of measure used for width. Auto (width value is ignored), pixels and percentage.
- **Display SubTotal label**: Displays the Total label or the Value itself.
- **Display line number**: Display the sequence number of the record in the Summary.
- **Display the hover on the Summary lines**: Apply the hover attribute by hovering the mouse cursor over the Summary lines.
- **Short value**: Displays the value abbreviated in the summary.
- **Display icon only on mouse over**: Displays the sorting icon only when the mouse is on the field label.
- **Fixed Label**: Establishes the column label in the screen top during the page scrolling.
- **Fixed columns**: This option allows that the Group By columns displayed on the Y-axis of the summary to be fixed during horizontal scrolling.
- **Refresh Interval**: Page refresh interval in seconds. If zero is provided there will be no refreshing.

Layout Settings

You can define the layout of the Summary for each type of Group By.



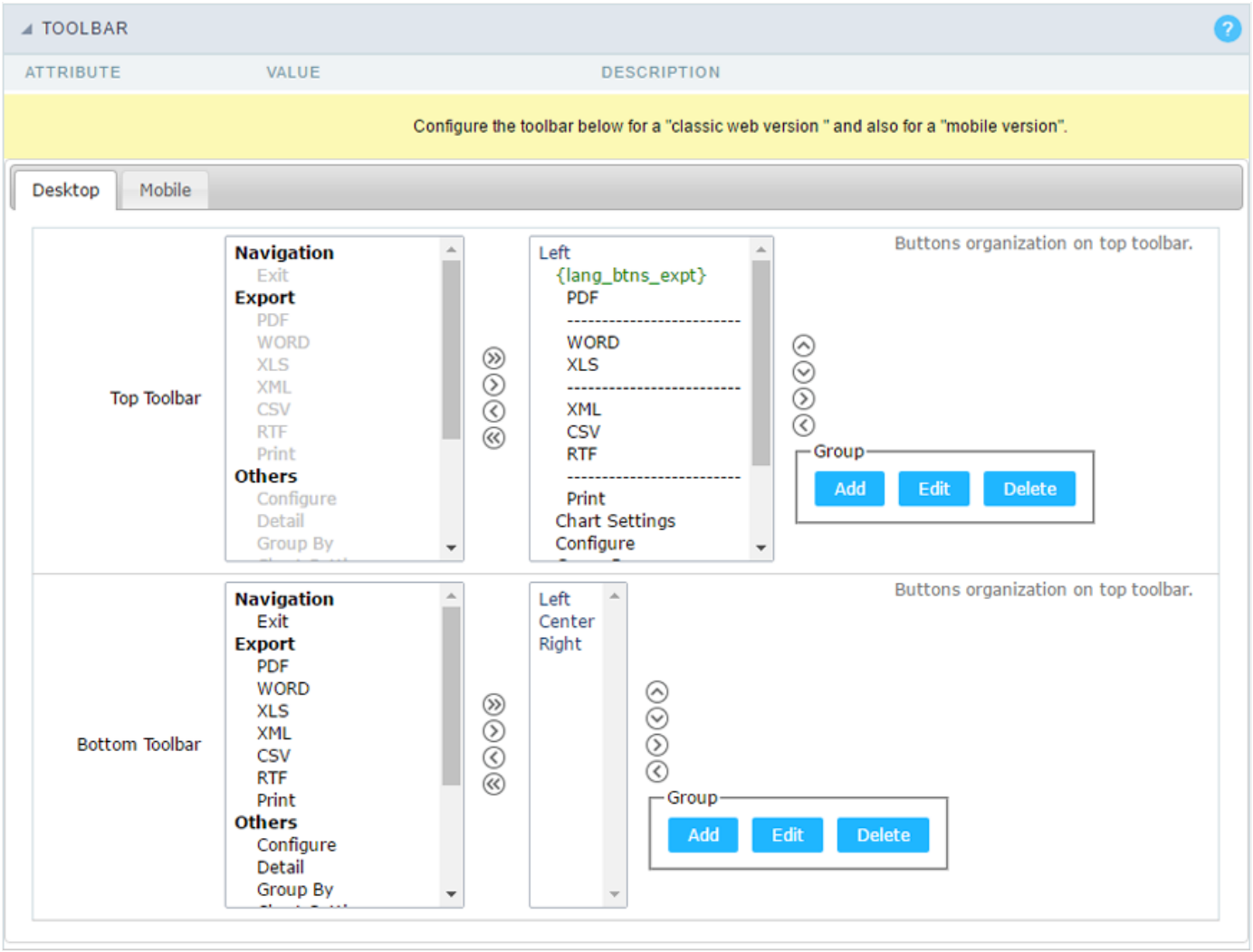
- **Group By Field**: Group By fields selected in the Grid.
- **Position**: Defines whether to use the x-axis or y-axis position.
- **Sorting**: Sets the sorting by the database value or by the display value.
- **Fill Empty Labels**: Defines whether the empty labels will be filled.
- **Link Grid**: Creates a link in the selected field.
- **Alignment**: Sets the layout alignment type to center, left, or right.
- **Tabular format**: Sets the Summary to the tabular format.

Toolbar

The toolbar is divided into two parts, top and bottom, so you can define which buttons will be displayed in both places. Button selection works independently.

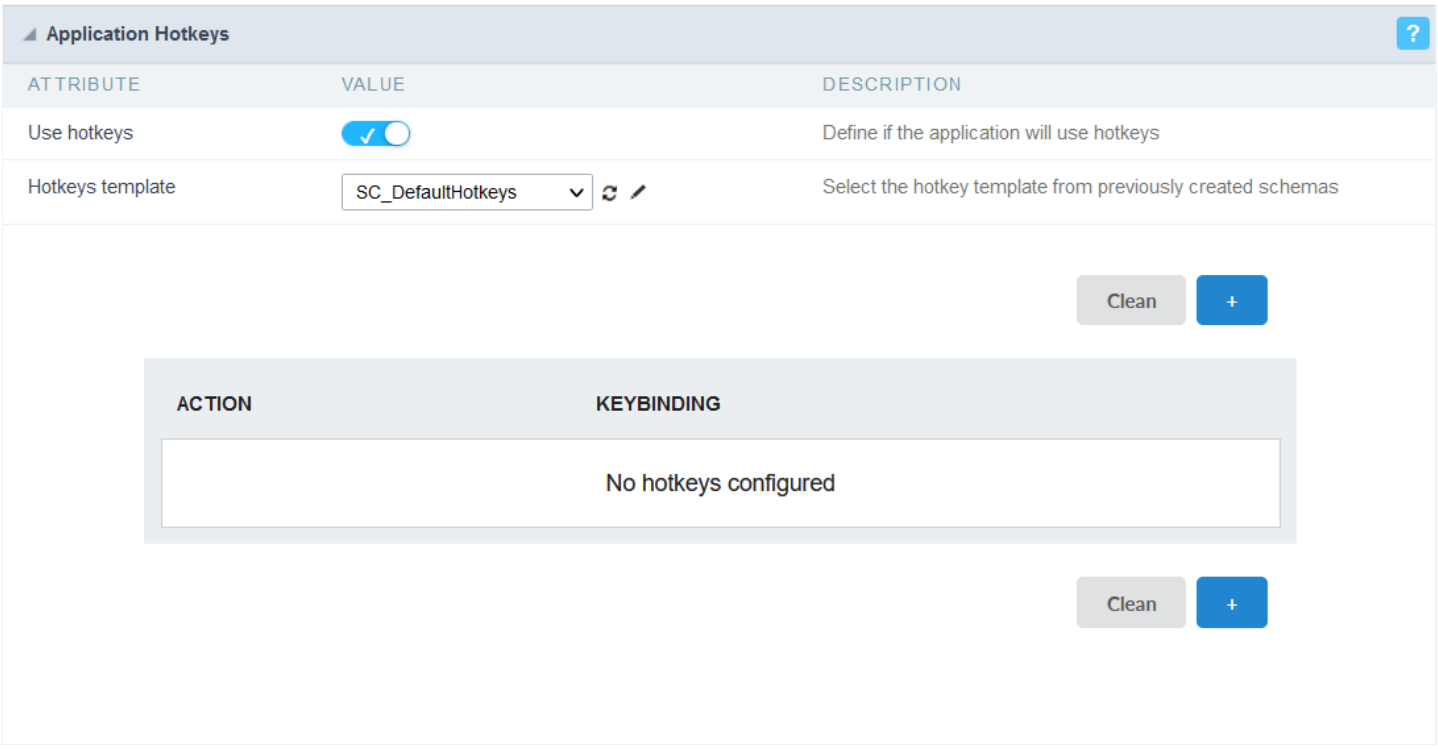
You can also define which buttons will be displayed when the application is accessed by a mobile device, just access the Mobile tab.

For more information about the toolbar, access Application > Grid > Toolbar.



Application Hotkeys

Scriptcase allows creating shortcut keys to your applications. You can select a predefined template or create specific actions for an application.



Use hotkeys

Defines if the application uses hotkeys. When you enable this option, the default shortcut keys settings are disabled.

Hotkeys templat

Select the hotkey template previously created.

Action

Selects the triggered action when pressing the selected key.

Keybinding

Selects the keys responsible for executing the chosen action.

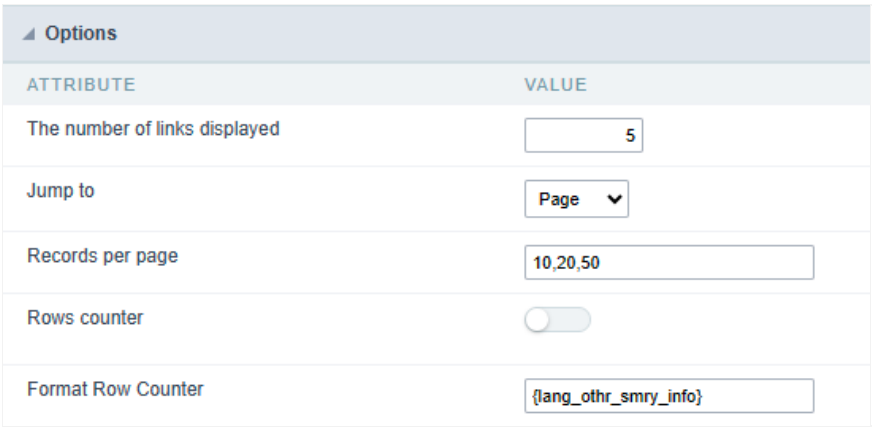
Add “+”

Adds a new action on the keys list.

Clear

It clears the selected hotkeys preference.

Options



The number of links displayed

Allows you to define the number of page links when the Page Navigation option is enabled.

Jump to

Allows you to define whether the redirection will be to a page or record.

Records per page

Allows you to define the options for the number of lines displayed in the combobox.

Rows counter

Shows the sequential number of rows in a partition of a result set. To use it, you must remove the row counter button from the toolbar.

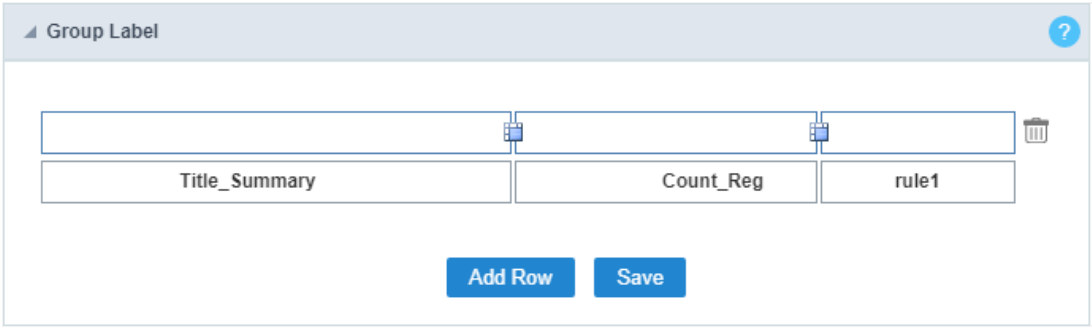
Format Row Counter

Allows you to define the format in which the counter lines will be displayed.

Example: [1 to 10 of 200].

Group Label

Click on Add Row, so that the group label can be added.

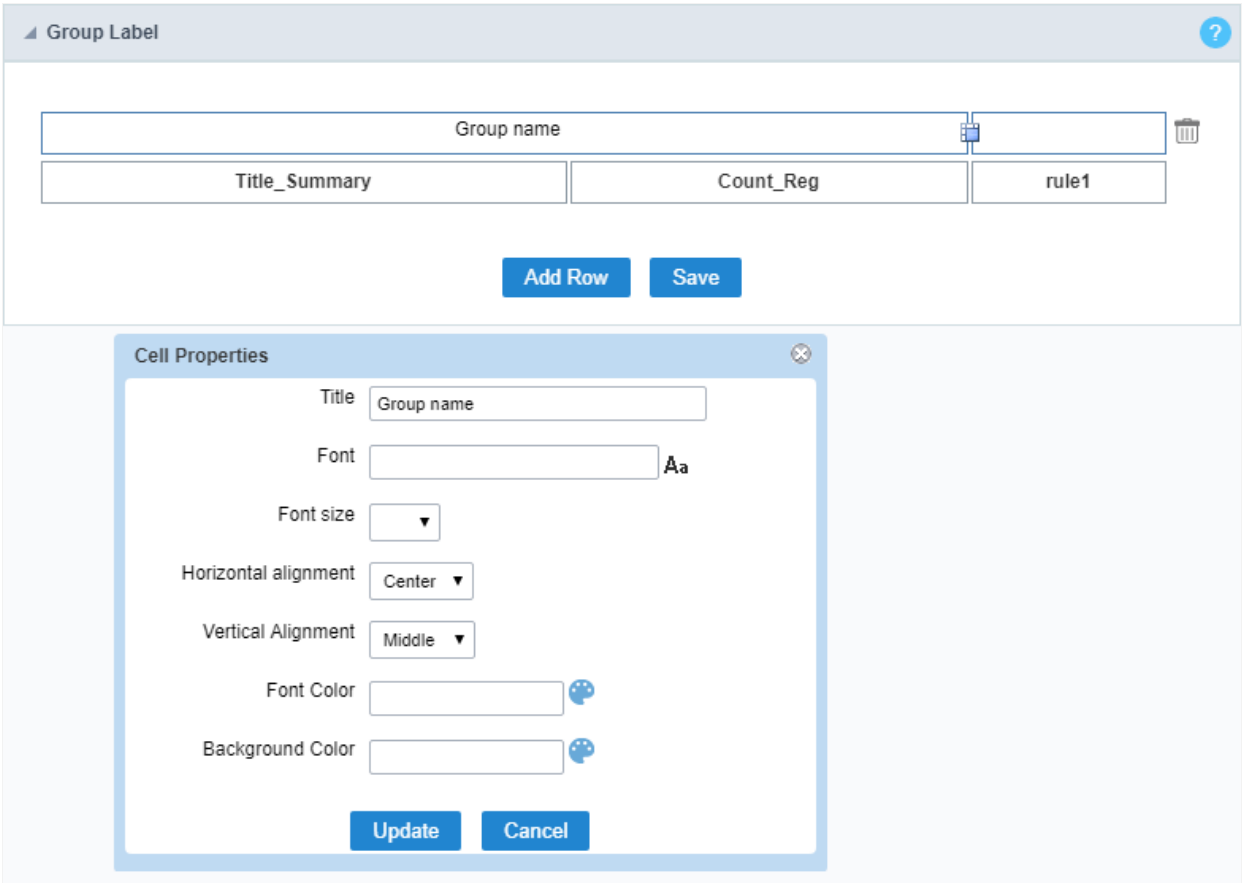


With this option, you can enter one or more titles for the columns of your applications.

If you have dynamic display fields, Group Label will not work.

To merge two or more cells, click the icon positioned between them.

To add a title, click the pencil icon to add a title.

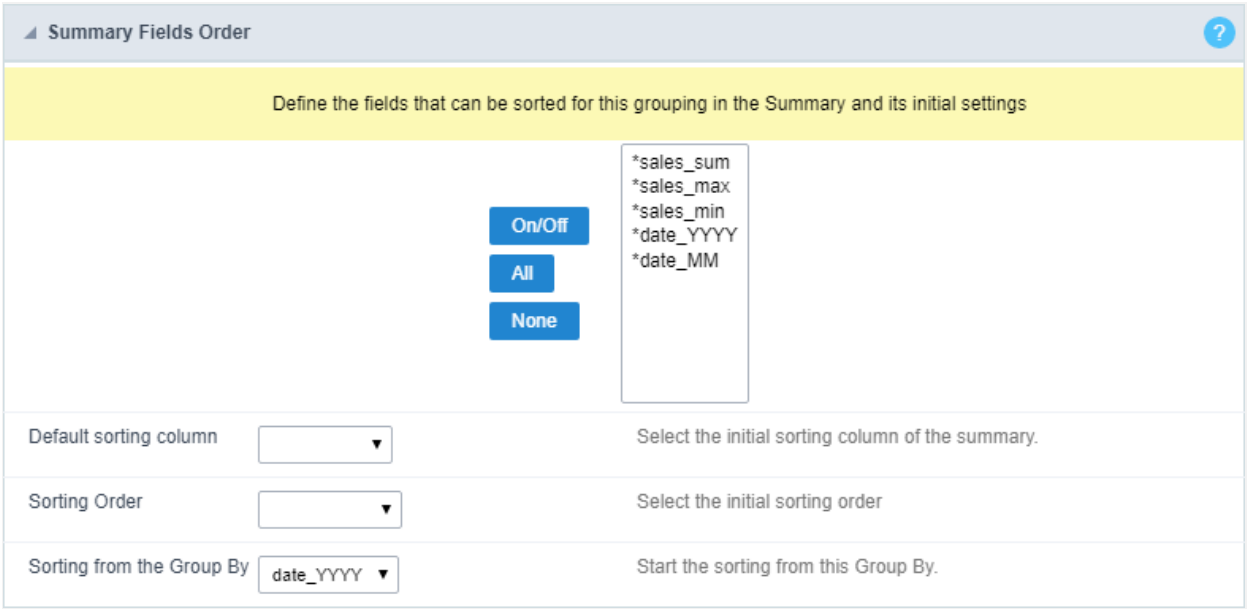


Title	Sets the title that will be displayed in your Group Label.
Font	Sets the font type of the title of your group label.
Font Size	Sets the font size of the title of your group label.
Horizontal alignment	Sets the horizontal alignment of the title of your group label. Being left, center or right.
Vertical alignment	Sets the horizontal alignment of the title of your group label. Being middle, low or top.
Font Color	Sets the color of the title text of your group label.
Background Color	Sets the background color of the space where the title of your group label will be displayed.

Sorting

You can define the fields in which you wish to allow the sorting, when running the application.

This configuration must be performed in each of the application group by.

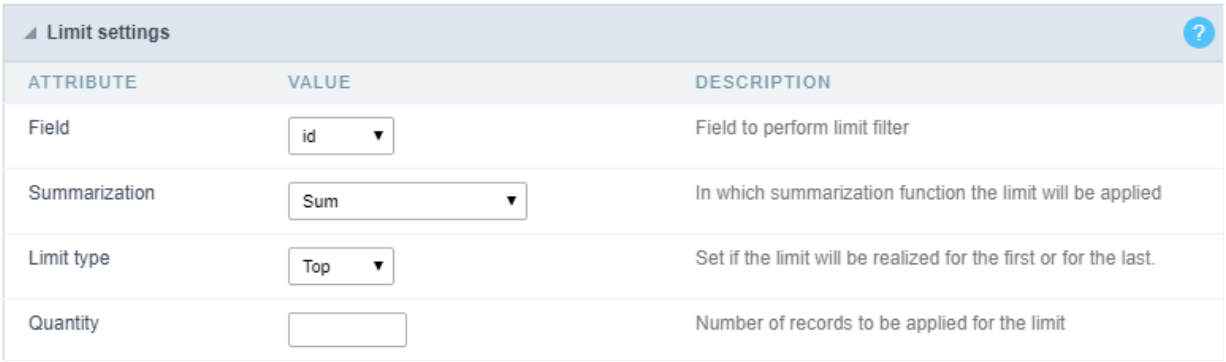


- **Fields**: Select the fields that you want to allow sorting.
- **Default sorting column**: Select a field for initial sorting in the Summary.
- **Sorting Order**: Choose whether the sort will be ascending or descending.
- **Sorting from the Group By**: Select one of the Group By for initial sorting.

Limit

Lets you limit the amount of records to be retrieved from SQL and displayed in the Summary.

This option is only available for static group by.



- **Field**: Defines the field that will be used to perform the limit.
- **Summarization**: Defines the summarization function in which the limit will be performed.
- **Limit type**: Sets the sorting that the limit will be applied, DESC or ASC.
- **Quantity**: Sets the amount of records that will be returned.

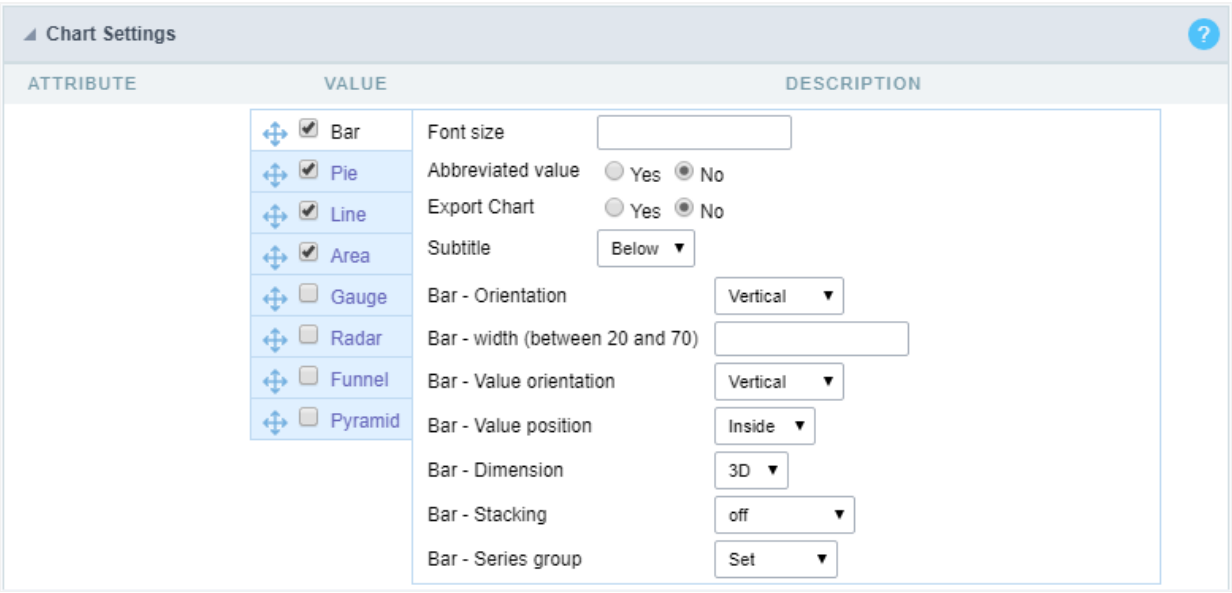
Charts

One of the modules of the Grid application is the charts, which in turn are generated based on Grid summary information.

Scriptcase charts are generated in HTML5.

Chart Settings

In the settings screen, you can edit the settings of all available charts.



To edit the specific settings for each Chart, select the Chart you want to edit.

To define the available charts when running the application, check the box next to the name of the Charts.

Below you will detail the specific settings of each of the Charts.

Bars

- **Font Size**: Sets the font size in the Chart. Enter only the font size, for example: 15.
- **Abbreviated value**: Defines whether or not the values displayed in the Chart should be abbreviated.
- **Subtitled**: Sets the subtitled position of the Chart.
- **Bar - Orientation**: Orientation of the vertical or horizontal bars on the Chart.
- **Bar - width (between 20 and 70)**: You can set the width from 20 to 70.
- **Bar - Value orientation**: The orientation of the Chart is defined and can be placed vertically.
- **Bar - Value position**: You can choose where the value will position itself.
- **Bar - Dimension**: Bar Dimension (2D or 3D) on the chart.
- **Bar - Stacking**: Stack the Bars in a single Bar (In series).
- **Bar - Series group**: Group the Bar chart in series.

Pie

- **Font Size**: Sets the font size in the Chart. Enter only the font size, for example: 15.
- **Abbreviated value**: Defines whether or not the values displayed in the Chart should be abbreviated.
- **Subtitled**: Sets the subtitled position of the Chart.
- **Pie - Format**: Pie or Donut format.
- **Pie - Dimension**: Dimension of the Pie Chart.
- **Pie - Sorting**: Ordering the Pie Chart.
- **Pie - Values format**: Formatting the displayed data.

Line

- **Font Size**: Sets the font size in the Chart. Enter only the font size, for example: 15.
- **Abbreviated value**: Defines whether or not the values displayed in the Chart should be abbreviated.
- **Subtitled**: Sets the subtitled position of the Chart.

- **Line - Shape**: Line Format
- **Line - Series group**: Line grouping type.

Area

- **Font Size**: Sets the font size in the Chart. Enter only the font size, for example: 15.
- **Abbreviated value**: Defines whether or not the values displayed in the Chart should be abbreviated.
- **Subtitled**: Sets the subtitled position of the Chart.
- **Area - Shape**: Format of the area displayed on the Chart.
- **Area - Stacking**: Stack the areas on the Chart.
- **Area - Series group**: Group the Chart in series.

Gauge

- **Font Size**: Sets the font size in the Chart. Enter only the font size, for example: 15.
- **Abbreviated value**: Defines whether or not the values displayed in the Chart should be abbreviated.
- **Subtitled**: Sets the subtitled position of the Chart.
- **Gauge - Shape**: Display format of the Chart.

Radar

- **Font Size**: Sets the font size in the Chart. Enter only the font size, for example: 15.
- **Abbreviated value**: Defines whether or not the values displayed in the Chart should be abbreviated.
- **Subtitled**: Sets the subtitled position of the Chart.

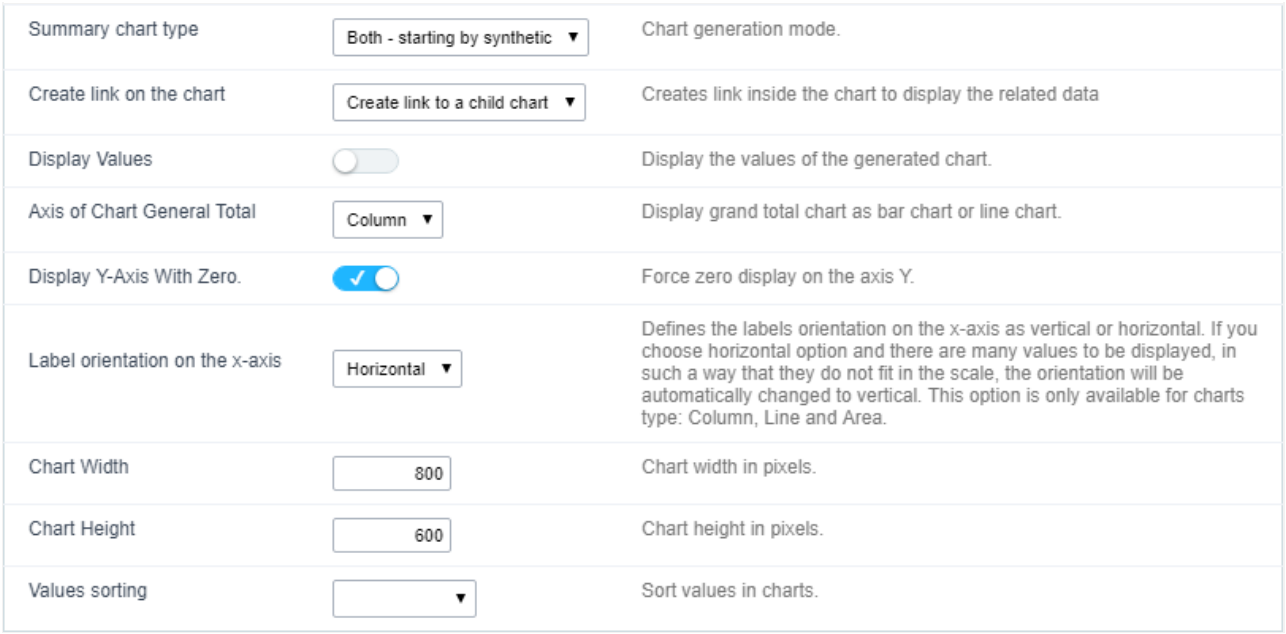
Funnel

- **Font Size**: Sets the font size in the Chart. Enter only the font size, for example: 15.
- **Abbreviated value**: Defines whether or not the values displayed in the Chart should be abbreviated.
- **Subtitled**: Sets the subtitled position of the Chart.
- **Funnel - Dimension**: Funnel dimension (2D or 3D) on the chart.

Pyramid

- **Font Size**: Sets the font size in the Chart. Enter only the font size, for example: 15.
- **Abbreviated value**: Defines whether or not the values displayed in the Chart should be abbreviated.
- **Subtitled**: Sets the subtitled position of the Chart. **Pyramid - Dimension**: Dimension (2D or 3D) on the Chart. **Pyramid - Values format**: Formatting the displayed data. **Pyramid - Sliced**: Display format of the Chart.

Below are the settings common to all Chart types.

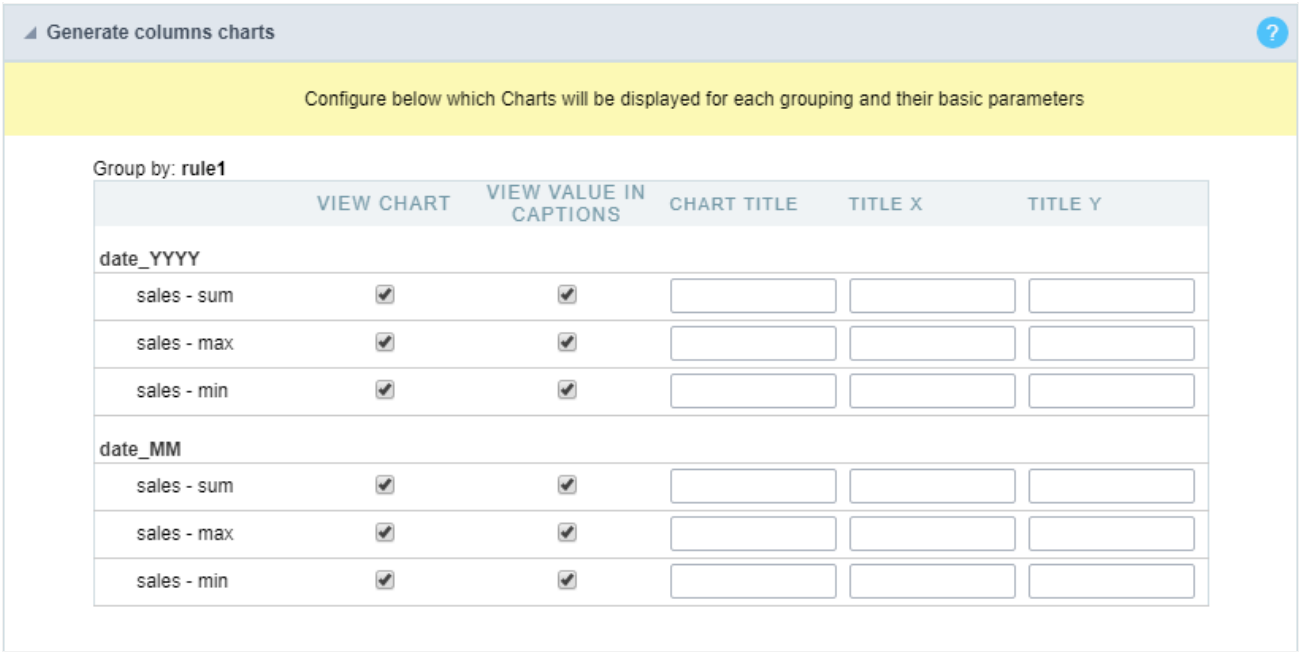


- **Summary chart type**: Defines how the Charts will be generated: analytical, synthetic, or both.
  - **Synthetic** They should restrict themselves to the first Group By the condition and allow the user to use the links to see details of the next set of criteria, increasing the level of detail.
  - **Analytical** They appear with the whole Group By criteria, creating a complete detailed Chart that does not allow the linking of other detailed Charts.
- **Create link on the chart**: Allows the Charts to have a link in their elements for detailed Grid applications or Charts. The data shown will be relative to the value clicked on the Chart.
- **Display Values**: Displays the values of the generated Chart.
- **Axis of Chart General Total**: Option to display the Chart of the grand total as column or row.
- **Display Y-Axis With Zero.**: Force display of the zero value on the Y axis.
- **Label orientation on the x-axis**: Sets the orientation of the labels on the X axis to vertical or horizontal. If the horizontal option is chosen and there are many values to be displayed so that they do not fit in the scale, the orientation will be automatically changed to vertical. This option is only available for Charts of type: Column, line and area.
- **Chart Width**: Width in pixels of the generated Charts.
- **Chart Height**: Height in pixels of the Charts generated.
- **Values sorting**: Sort the values of the Charts.

Column Charts

You can define the display of the column Charts as well as the label for the Chart and the axes. These settings are set individually for each Group By.

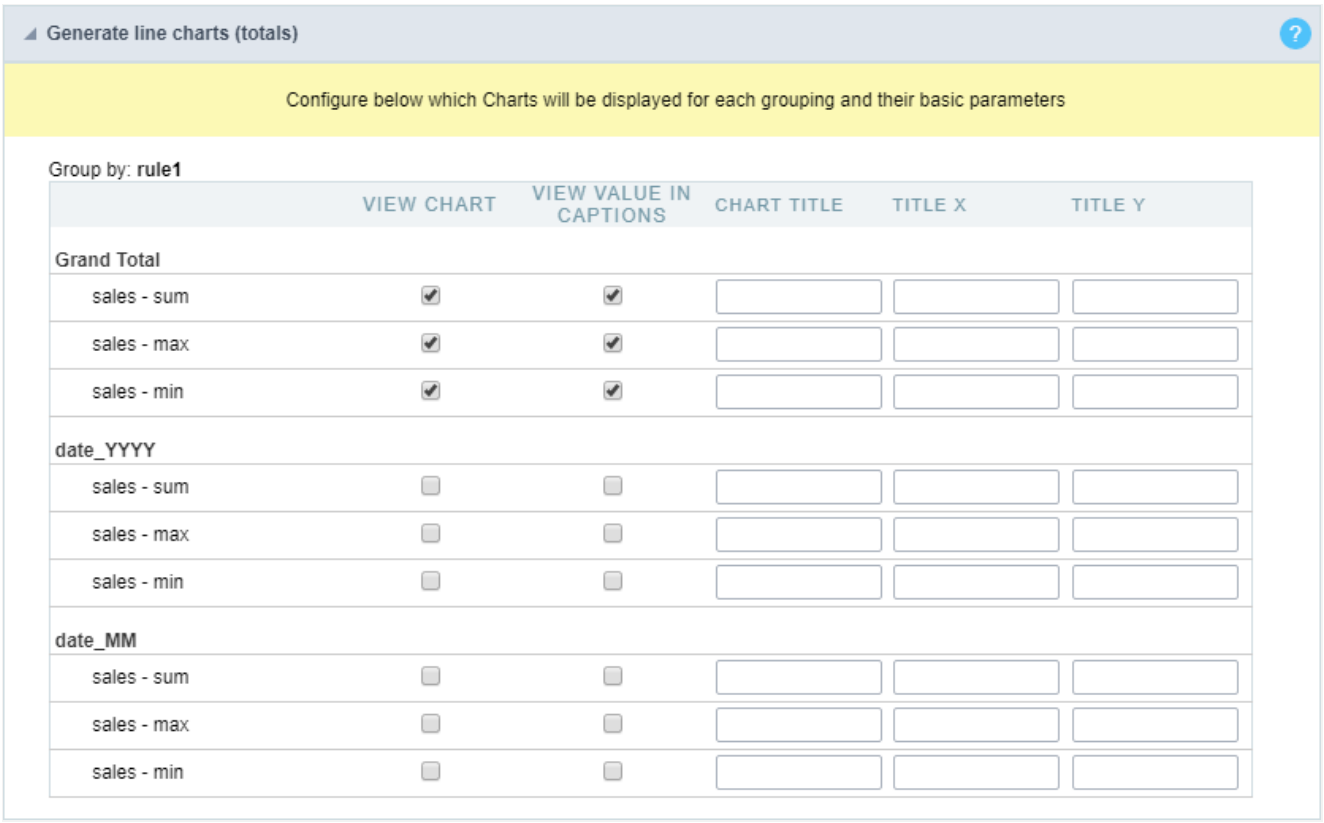
If the user sets up more than one field, the analytical chart will only generate with the first two.



Line charts (totals)

The use of line Chart (totals) configuration is only available when you have a Group By with at least two fields.

To configure this option, go to the “Group By” menu and create a static Group By some fields or add fields to dynamic Group By.



Layout

The Chart theme tool enables you to fully edit Chart themes.

These themes can be set only for the current application, when changed in the application itself or for all Charts when these settings are performed in the Charts editor.

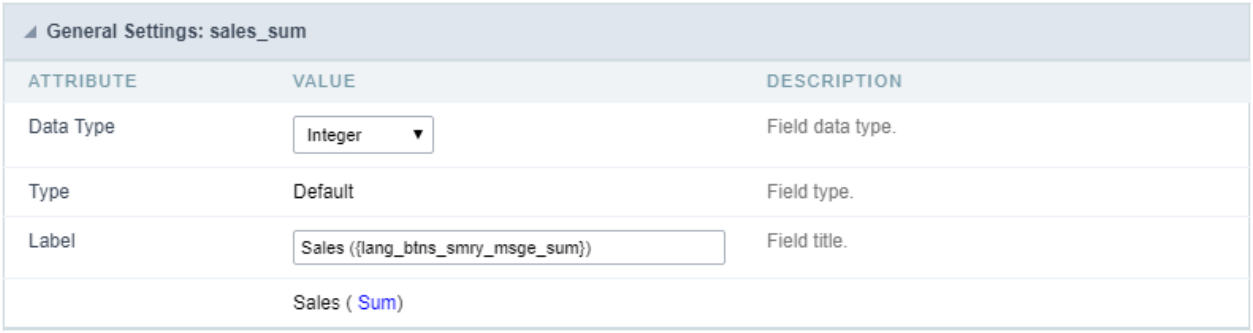
For more information, access Layout > Chart Themes.

Fields

All fields added in the Summary totals, in the dynamic Group By or in the static Group By, will be listed here separated by the Group By name.

General Settings

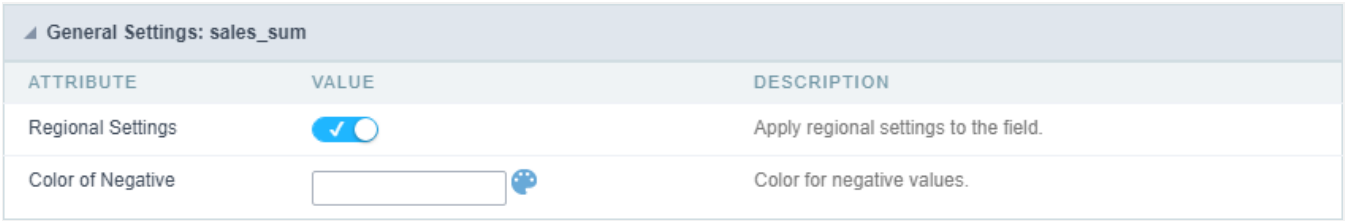
General settings of total fields in the Summary.



- **Data Type**: Defines the data type for the field.
- **Label**: Defines the label for the displayed field in the Summary.

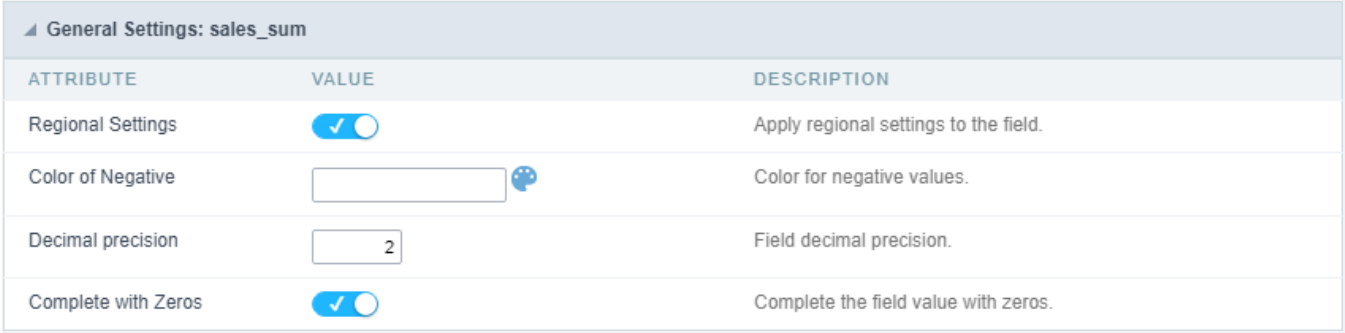
Values Formatting

Integer Type



- **Regional Settings**: When you activate it you apply the regional settings for this field. To configure them, click on the menu Locales > Regional Settings.
- **Color of Negative**: In this field you can enter a color in hexadecimal. (Example: # 000000)

Decimal Type



- **Regional Settings**: When you activate it you apply the regional settings for this field. To configure them, click on the menu

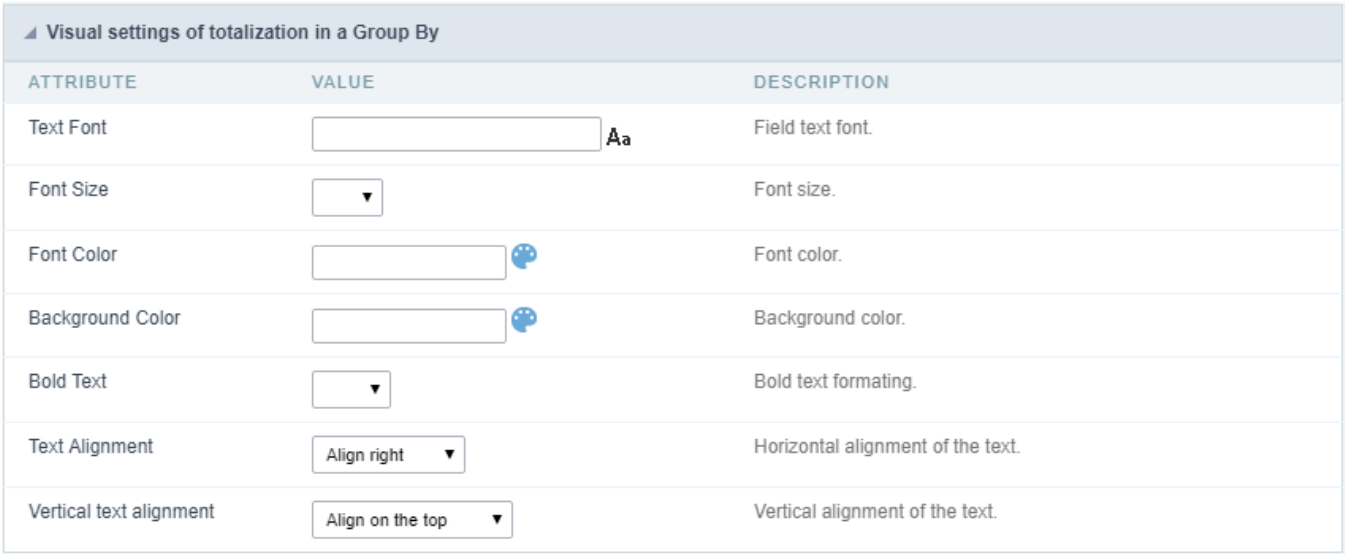
Locales > Regional Settings.

- **Color of Negative**: In this field you can enter a color in hexadecimal. (Example: # 000000)
- **Decimal Precision**: Number of decimal places your field will have on display.
- **Complete with Zeros**: Enables the field value to be filled with zeros.

Currency Type

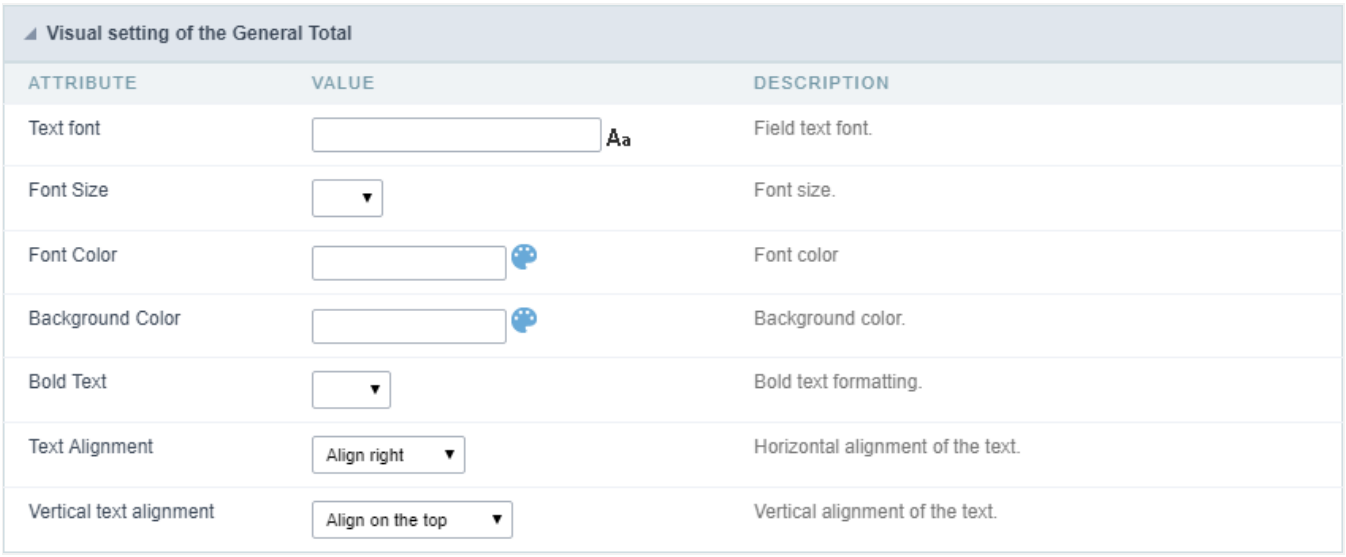
- **Regional Settings**: When you activate it you apply the regional settings for this field. To configure them, click on the menu Locales > Regional Settings.
- **Currency Format**: When you enable this option the application will display the currency symbol according to the regional setting.
- **Color of Negative**: In this field you can enter a color in hexadecimal. (Example: # 000000)
- **Decimal Precision**: Number of decimal places your field will have on display.
- **Complete with Zeros**: Enables the field value to be filled with zeros.

Visual Settings of Totals in Group By



- **Font** : Allows you to define the font used in the Group By label.
- **Font Size** : Allows you to set the font size used in the Group By label.
- **Font Color** : Label color in Group By.
- **Background Color** : Group By background color.
- **Bold Text** : Enables or disables the Group By label text in bold.
- **Text Alignment**: Sets the horizontal positioning of text.
- **Vertical text alignment**: Sets the vertical positioning of text.

Visual Settings of The General Total

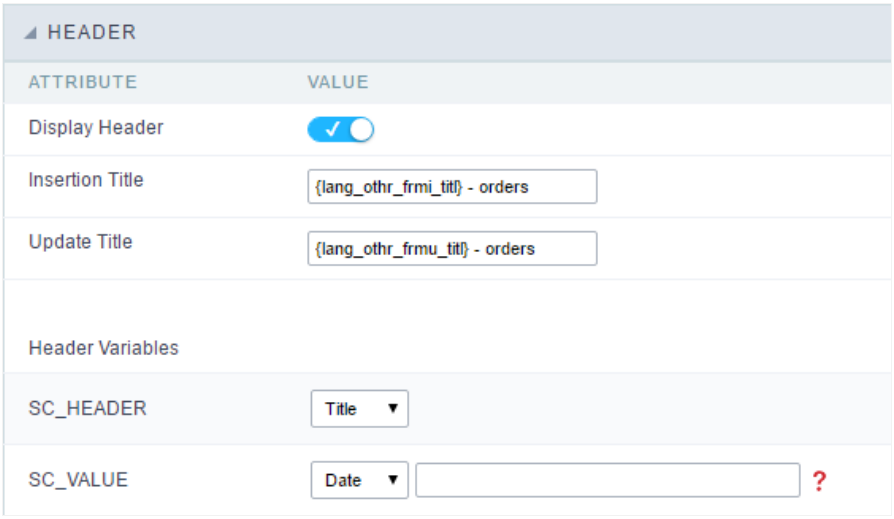


- **Font** : Allows you to define the font used in the Group By label.
- **Font Size** : Allows you to set the font size used in the Group By label.
- **Font Color** : Label color in Group By.
- **Background Color** : Group By background color.
- **Bold Text** : Enables or disables the Group By label text in bold.
- **Text Alignment**: Sets the horizontal positioning of text.
- **Vertical text alignment**: Sets the vertical positioning of text.

Layout

Header

In this configuration, you can define the information that will be displayed in the header and footer of the Summary.



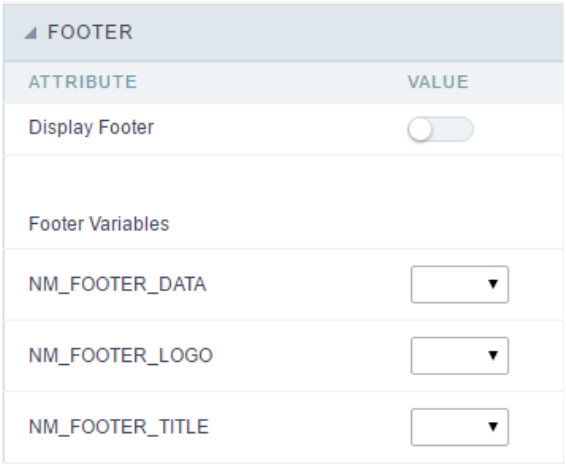
This screen may vary depending on the header format you choose within the Display folder.

- **Display Header**: This flag determines whether the header will be displayed.
- **Summary Title**: Lets you enter a title to be displayed in the application.

- **Variables**: Variable fields can be filled with any of the types displayed in the Combobox, depending on the type you need to associate a content with the required one. Below you describe the existing types:

- **Field**: When the "**Field**" option is chosen, it will open a Combobox next to the fields that are part of the "Select". By choosing one of these fields, you are associating the value of the field to display in the Header or Footer.
  - **Date** : When "**Date**" type is selected, the system date in mm/dd/yyyy format will be displayed in the Application Header or Footer. There are several display formats using the date and time of the server. The format can be entered in the text field next to it. To access existing formats click on and an explanatory window will appear.
  - **Image** : When the image type field is selected, a field for filling in the image name on the server appears. To locate the existing images and select one, click on the "**Choose Image**" icon and to make new images available on the server click on "**Upload**" .
  - **Value**: When "**Value**" type is selected, the content that is filled in the text field next to option, appears in the Header or Footer. You can enter texts and "**Global variables**". Ex: "Employee Name: [v_name]".

Footer



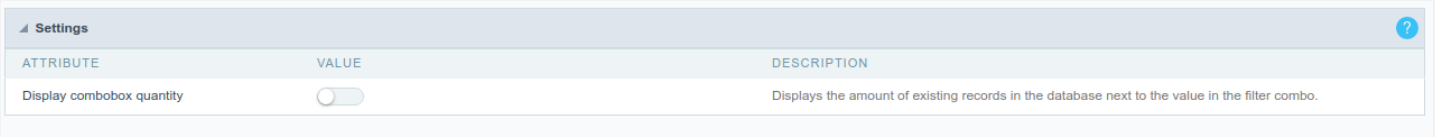
This screen may vary depending on the Footer format you choose within the Display folder.

- **Display Footer**: This flag determines whether the Footer will be displayed in the application.
- **Variables**: Variable fields can be filled with any of the types displayed in the Combobox, depending on the type you need to associate a content with the required one. Below you describe the existing types:

  - **Field**: When the "**Field**" option is chosen, it will open a Combobox next to the fields that are part of the "Select". By choosing one of these fields, you are associating the value of the field to display in the Header or Footer.
    - **Date** : When "**Date**" type is selected, the system date in mm/dd/yyyy format will be displayed in the Application Header or Footer. There are several display formats using the date and time of the server. The format can be entered in the text field next to it. To access existing formats click on and an explanatory window will appear.
    - **Image** : When the image type field is selected, a field for filling in the image name on the server appears. To locate the existing images and select one, click on the "**Choose Image**" icon and to make new images available on the server click on "**Upload**" .
    - **Value**: When "**Value**" type is selected, the content that is filled in the text field next to option, appears in the Header or Footer. You can enter texts and "**Global variables**". Ex: "Employee Name: [v_name]".

Search

Settings



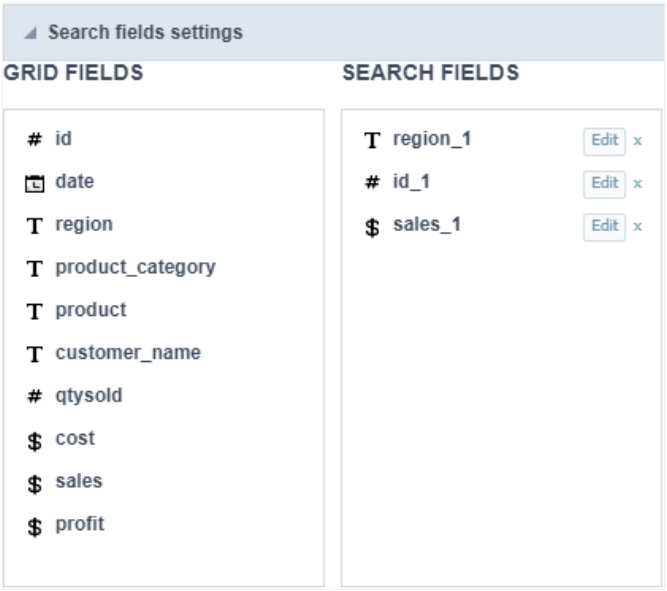
- **Display combo box quantity** : Allow to display the quantity of existence records in the table net to the value in search combo.

Settings of search fields

To define the fields that will be used in the Summary search, use drag-and-drop to position the fields within Search Fields.

The order of the fields, within Search Fields, will be the display order in the generated application.

This search works only in Grid summary.



Editing Fields

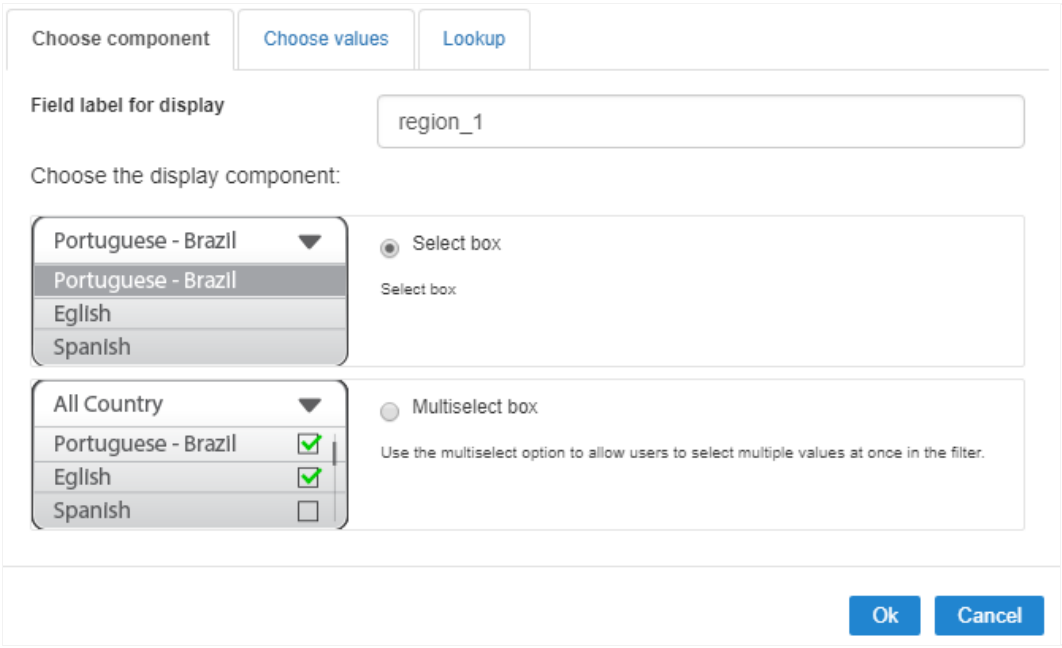
The same field can be added more than once to the Search, these fields will be visible in the generated application since it uses distinct settings, if they are using the same configuration only one of the fields will be displayed.

The display settings of the fields are performed individually, and can be accessed by adding the field in the search and clicking edit.

you will separate the explanation of the display settings from the fields according to the types.

Types: Text/Special

Configuration interface for text and special fields.



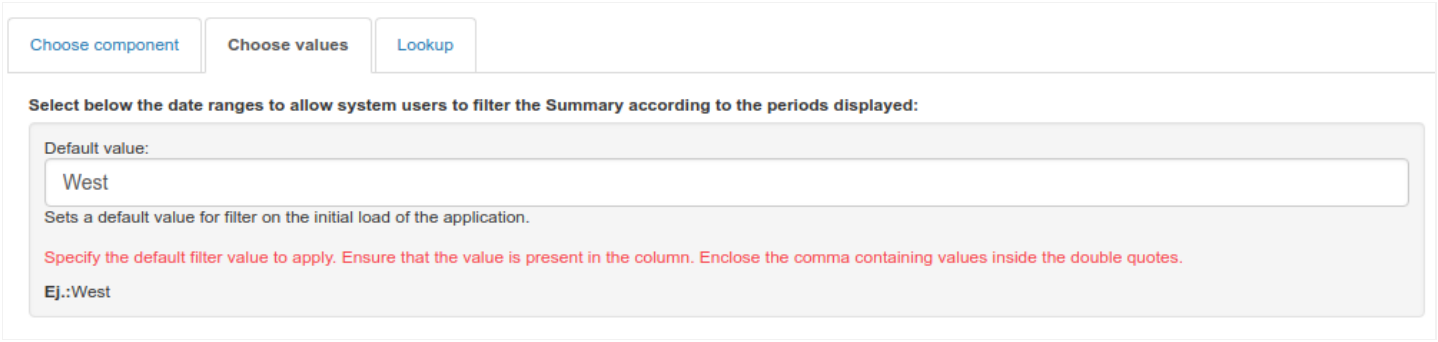
Choose Component

You must define how the field will be used in the Search. Each field type, text, number, and date have different configuration options.

- **Search field label** : Defines the label of the field that will be displayed in the Search.
- **Choose component type**: It defines the format of use of the fields in the Search, for text fields you have the Select Box and the Multi select box.

Choose values

The **default value** field allow us to define a default value to the summary search in the initial application charge.

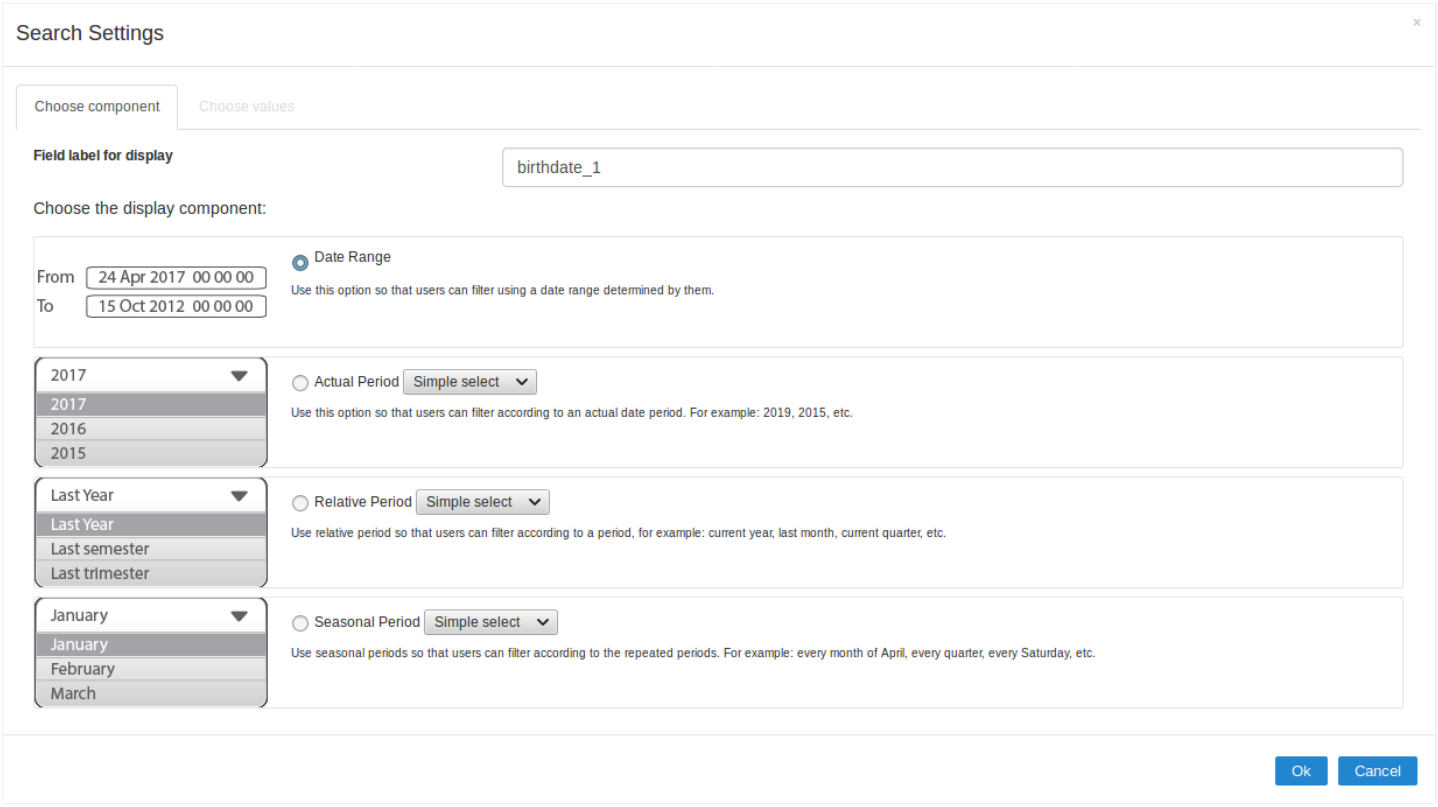


Lookup

In this tab, you can configure the display lookup in the Search field. For more information on creating a view lookup, access **Application > Grid > Fields > Text**.

Types: Date/Time

Configuration interface for date and date time fields.



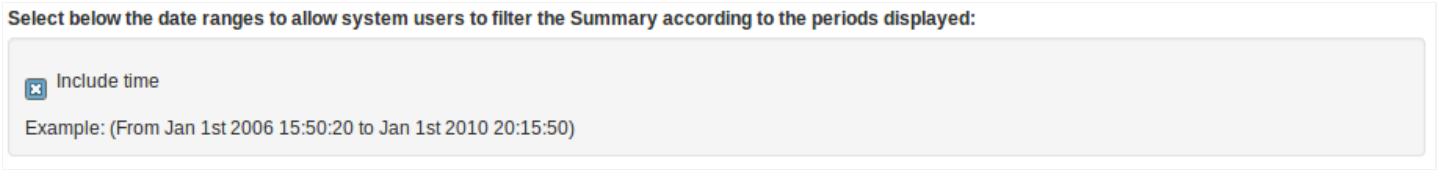
Choose Component

You must define how the field will be used in the Search. Each field type, text, number, and date have different configuration options.

- **Search field label** : Defines the label of the field that will be displayed in the Search.
- **Choose component type**: Defines the format of use of the fields in the Search, for fields date and datetime, you will have the following options: Date Range, Actual Period, Relative Period and Seasonal Period.

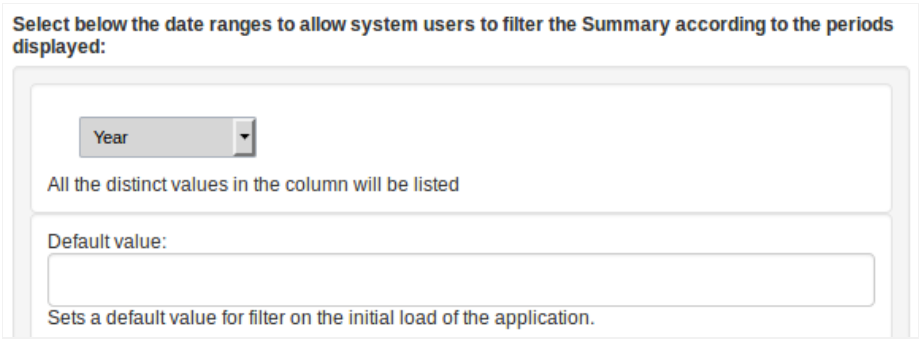
Choose Values - Date Range

For datetime fields, you must enable the option **Include Time** so that the hours can be included in the search



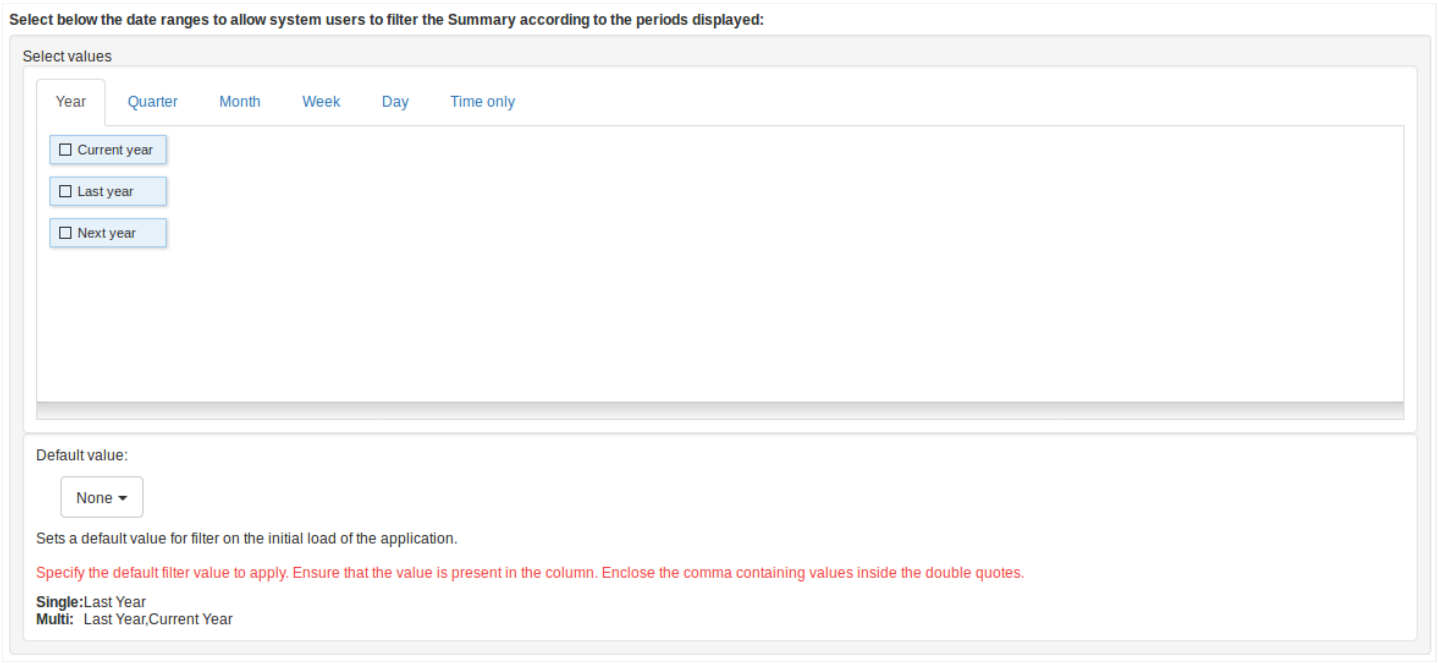
Choose Values - Actual Period

You must define the period that will be used in the search. When running the application, you will have a select with the dates displayed according to the selected period.



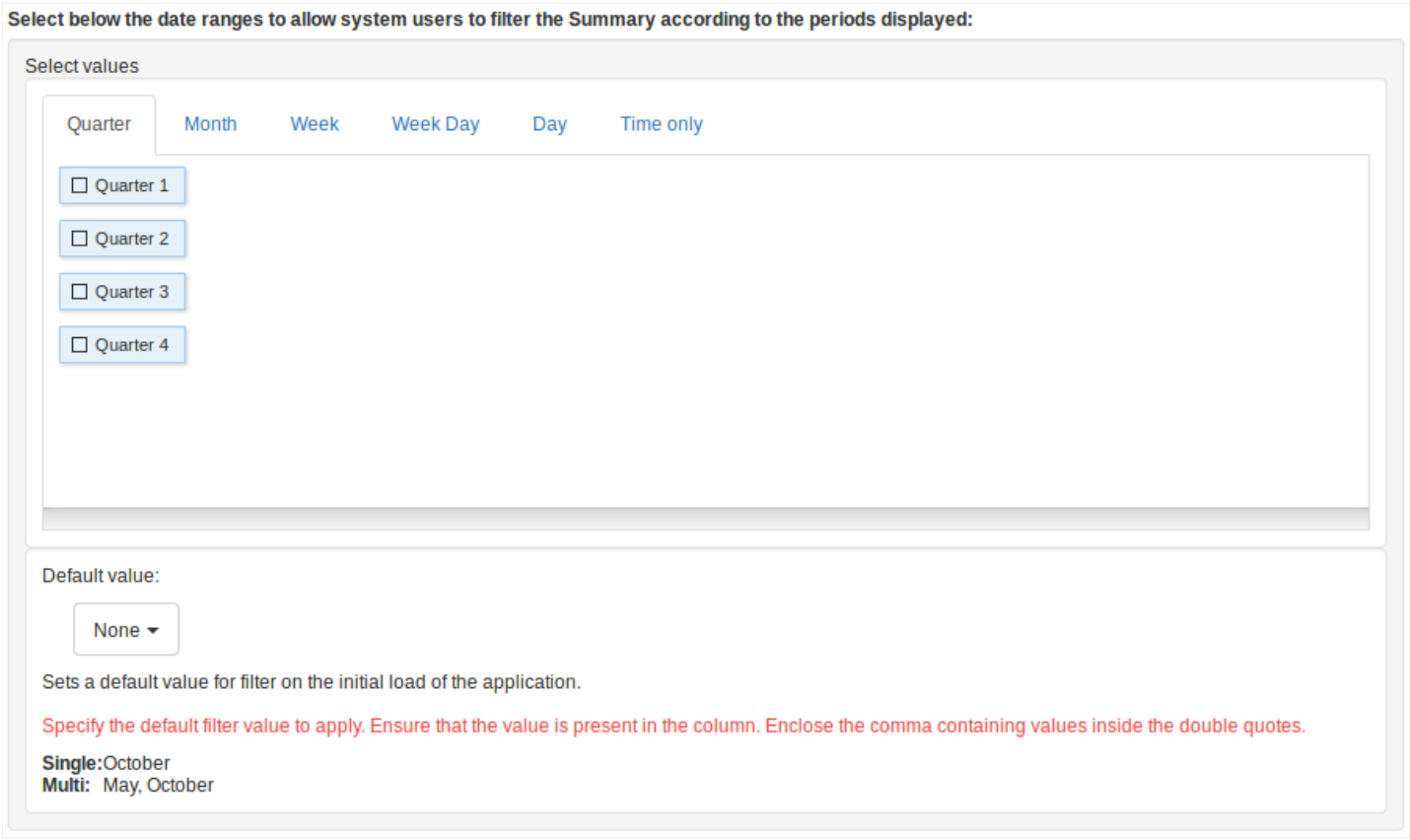
Choose Values - Relative Period

You must define which periods are available for use in the search.



Choosing Values - Seasonal Period

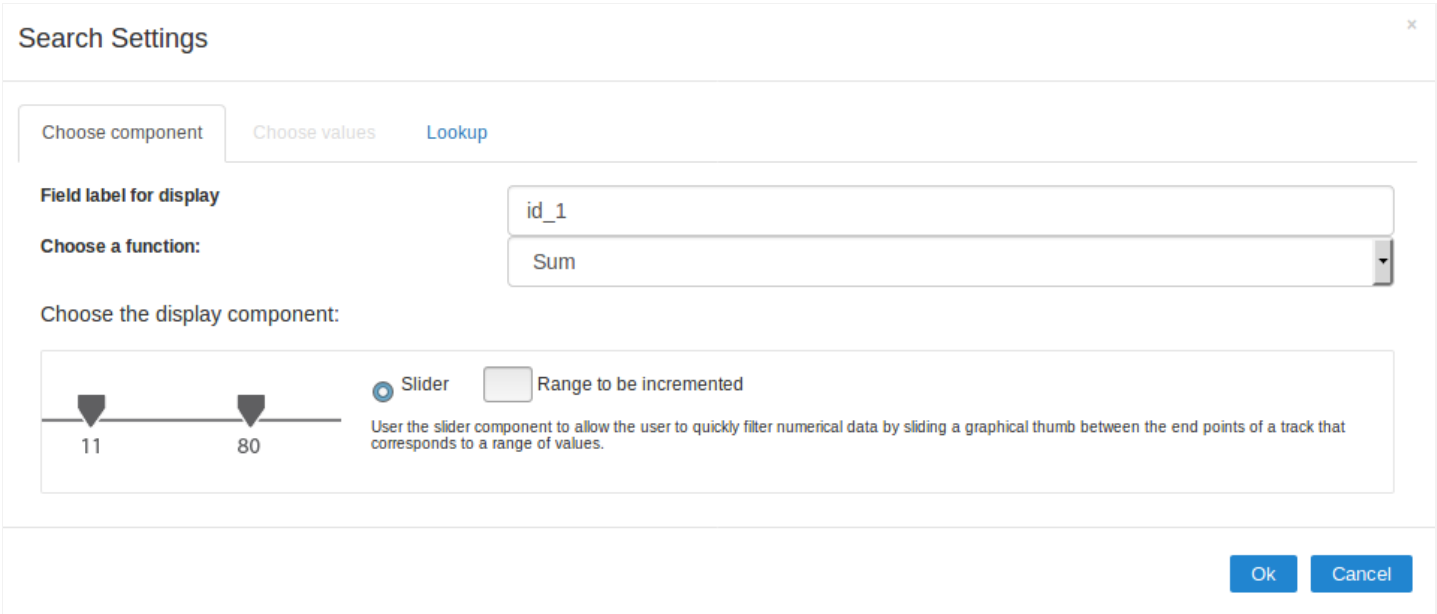
The available values are separated by tabs:



- **Quarter**: Defines the quarters used in the search.
- **Month**: Defines the usage of the months of the year.
- **Week**: Defines the usage of the weeks of the year.
- **Week Day**: Defines the use of the days of the week.
- **Day**: Defines the usage of the days of the month.
- **Only Time**: Defines the use of the day time (Available only in the datetime field)

Types: Number

Configuration interface for numeric fields.



Choose Component

You must define how the field will be used in the Search. Each field type, text, number, and date have different configuration options.

- **Search field label** : Defines the label of the field that will be displayed in the Search.
- **Choose function to apply**: Defines the function that will be used in the search for setting the value. For example, when choosing sum, the range is the range between the lowest and the result of the sum greater.
    - **Actual Values**: This option sets the display of the actual value of the field, saved in the database, without the use of any of the aggregate functions. When using this option, two types of searches, select box and Multi select box usage are added, as well as enabling lookup settings for numeric fields.
- **Choose component type**: Defines the format used for the fields in the search, for numeric fields, you have the Range and selecting **Actual Values** will be displayed **Select Box** and **Multi select box**.

Lookup

For numeric fields, this option is available only when you use the **Actual Values** option and the way the field is used in the search is **Select box** or **Multi select box**.

In this tab, you can create a display lookup in the search field. For more information about creating view lookup, go to **Application > Grid > Fields > Integer**.

Related Links 🔗

Related Video ▷

Grid Search Settings

With this interface, you can define general options of the Search Form.

Search configuration Interface.

Search Criteria

Allows to select the logical operator **AND** or **OR** to define the criteria of the search;

Display Condition

Gets the condition of the search available for the user to choose one. He can select “AND” or “OR” in a Combobox.

Use auto-complete in the fields

Automatically turns the field into an autocomplete according to the existing values in the database. If the user chooses **Yes**, the autocomplete will enable automatically in all inputs that contain a relationship. If the user decides **No**, so no autocompletes will be displayed. Otherwise, the option selected is **Defined in the field** it’ll keep the settings for each field individually.

Related Links 🔗

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Ads
- Search with Advanced and Dynamic search
- How to Add a Pivot Table with Filter
- Pivot Table with Filter
- Highlight option

Related Videos ▷

Grid Search Criteria Settings

With this interface, you can configure the conditions available for each field of the Search form.

Search configuration Interface.

We can see the fields list on the left combo. On the right, the list of options for filtering the selected field. To select an option, click on one of them (Equal to, Beginning with, Contains, etc.) and then the button On/Off. The arrows, on the right, allows altering the order of the fields.

For the Date type fields, you can define special conditions for the search, accessing the field configurations, and editing the Special Conditions Settings.

Below the list are the buttons to enable the selected options:

- **On/Off**: Enables or disables the field or the option chosen.
- **All**: Marks all fields or options.
- **None**: Unmarks all the fields or options.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refine search and Highlight option
- Search with blocks
- Quick Refined Advanced and Dynamic search
- Pivot Table with Filter
- Macro Sql_where_filter
- Highlight option

Related Videos

Advanced Search Settings

Through the table below we set all the options that will be part of the application Grid Search.

Horizontal Alignment

Allows you to define the Horizontal positioning of the Search Form.

Margins

Defines the position of the margins of the Search Form.

Keep Values

It keeps the searched values when the user returns to the search form.

Keep Columns and Order Selection

Set it to preserve the selected columns and sorting for each search, if they went changed by the user through the toolbar options.

Use Enter to

It allows you to define the action that the Enter Key has on the Search form. **Tabulate** enables you to navigate between fields, and **Submit** performs the search(activates the Search button).

Display Tags

Allows displaying as tags, the searches used for the Grid.

Display after filtering

Display tags only after performing an advanced search. If disabled, it will always display a tag, regardless of the advanced search.

Unify results

Sets the chars limit to group the result of the tags. This option should be used when the field type is multiple-select.

Treeview in the Tags

Sets the use of Treeview for tags.

Initial status of the Treeview

It sets the initial state of the Treeview. (If the app is using Treeview for tags)

Start open

It displays the tags.

Start close

It displays the full description of the tags as text.

Clear other filters after submit

After applying the advanced filter, all other filters will be deleted.

Interact with the dynamic filter

Apply the same filter in matching fields between Advanced and Dynamic Filter, when not showing the labels.

Highlight

Highlights the results in the query. It only works for “Exactly the same”, “Contains” and “Equal start” conditions.

To customize how the highlight will be displayed, you must access the theme editing tool .

1 - Access the **Layout > Application Themes** menu.
2 - Choose the desired theme and click **Advanced Mode**.
3 - Then look for the sub-item **Grid**, within the item of the same name.
4 - In this sub-item you will find the option “Line”, which contains the folders **Line Odd and Even** . In each of these folders you will find the option **Quicksearch highlight**, with the options for editing Text, Border and Background of the searched terms.

Related Links

- Search with buttons in line
- Search with a predefined initial value
- Search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos

Select Fields for Advanced Search in Grid

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

- - Move all fields to the right.
- - Moves only selected fields to the right
- - Moves only the selected fields to the left.
- - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the button.

> To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning

Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

> Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro se_where_filter

Related Videos

Imagens utilizadas no arquivo includes/docs/app/posicionamento_campos –> [img_posicionamento_campos]: /assets/images/docs/app/comum/campos/posicionamento_campos/posicionamento_campos_search.png

Required Fields For Advanced Search in Grid

Defines which fields of application will be required for the search.

The application generated will be displayed a bullet (*) next to the field and an error message is generated if not assigned no value.

- **Marker position** : Marker’s position relative to the field.
- **Display message** : Displays whether or not the validation error message.

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos

Grid Search Toolbar

Search Toolbar

The Search toolbar is divided in two parts: Top and Bottom, in a way that is possible to define the buttons that will be displayed in both bars. The selection of buttons in the top and bottom toolbar works independently, allowing the buttons to be displayed in both bars at the same time.

Toolbar Interface.

Navigation:

Groups the options relative to the navigation buttons that can be displayed in the application.

- ***Search***: Execute the search.
- ***Clean***: Clean the all the search fields.
- ***Edit***: Enable the ***Save Tag*** option.
- ***Exit***: Exit the application.

Others:

Groups a diversity of options relative to the application.

- ***Languages***: Displays a combobox with the names available, defined in the project properties.
- ***Themes***: Displays a combobox with the themes available, defined in the project properties.
- ***HelpCase***: Displays a button to redirect to the help page.

Separator

- **———————**: Dispays a line separating the buttons, when used the Group Buttons.

Use in-line buttons:

Allows the alignment of the filter buttons next to the fields.

Inline buttons: Allows you to select which buttons will be displayed next to the field, and you can sort them according to your wishes. This option is available by enabling **Use buttons inline** in the button settings.

When activating the **Inline Buttons** option, the screen to configure the buttons will be displayed.

> The buttons available in this in-line button option are the same as those shown in the standard toolbar described above.

This is the result when using a radio button on the line

Button Settings

Button Settings Interface.

- **Hotkey**: Allows you to set keyboard hotkeys to a button.
- **Position of the in-line buttons**: Sets the positioning of the buttons to the right or left of the fields.
- **Column Quantity**: Sets the number of display columns of the buttons, allowing you to configure whether they will be displayed side-by-side or distributed in columns.

Options ![Options Interface.][barra_ferramentas_filtro_opcoes] *Options Interface.* * __Button Position(Top/Bottom)__ Positioning the buttons of the toolbar Top/Bottom. -->

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value

Related Videos ▷

- [arch with relative periods](#)
- [Search with blocks](#)

- [Highlight option](#)
- [Macro sc_where_filter](#)

Save Search in Advanced Search (Grid)

This feature allows the end-user to save his searches in a profile. You can create some rules, like to save the searches by user login.

 Save Filter Interface.

Save Filter Interface.

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos

Grid’s Advanced Search Events Overview

In events, blocks can be used with global variables, local, JavaScript code, CSS codes and Scriptcase macros.

onScriptInit - Advanced Search - Grid

This event occurs just once before it display the search form. It can be used to set some default values to the fields. For example: {company} = [glo_company]; Date fields are used differently, They use a suffix "day" for day, "month" for month and/or "year" for year. For example, The {Birth} field will be: {Birth_day}=date("d"); {Birth_month}=date("m"); {Birth_year}=date("Y");

onRefresh - Advanced Search - Grid

onRefresh

This event occurs when the form application is reloaded, is possible to reload the form based on fields (select , radio, checkbox)

onSave - Advanced Search - Grid

This event occurs always that some search setting is saved in the filter application.

onValidade

This event occours when the search form is submitted. Ex. in order to prevent some searches or to validate any search params if(strlen({name}) <= 0) { sc_error_message(" Please fill the name field before submit your search "); }

Grid’s Advanced Search Layout Overview

On this module are available the features of editing, attributes and display of the application, in a way that you can apply display themes, organize blocks, define values and the format of the Header/Footer and etc.

Blocks

Conceptually a block is a “container” where you can position the fields of the Applications: Form, Control or Grid with Slide orientation.

By default, the applications created in ScriptCase are built with a only one block, with the same name as the application. You can add the amount of blocks that you want to organize in a more convenient way. The page below, observe that theirs a column Organization, and that is where you’ll define if the next block will be set beside or below the current one.

Application Block configuration

On the left side of each block there are two icons, first has the function to edit all the information relative to the block and the second is to delete the block.

- Organizing the position of the Blocks
  - See below how to modify the display order of the Blocks in one Page.

  Click and drag the block that you desire to modify to its new position.

  Application Block Display configuration

  - See how to remove a block from display

  Click on the block desired and drag it to the item “Blocks not Shown”. This way, you can also drag the block to another page if desired. See the images below.

  Application Block Display configuration

  Application Block Display configuration

  - **Attributes**
    - **Block**
      - **Name** : Name of the Block.
      - **Label** : Title of the block that’ll be displayed in the application.
    - **Title**
      - **Display** : Flag that controls the title display of the block.
    - **Label**
      - **Display** : Flag that controls id the label of the fields will be displayed in the block.
      - **Position** : Options to display label :
        - **Above** : The label will be displayed above the field.
        - **Beside** : The label will be displayed beside the field.
        - **Below** : The label will be displayed below the field.
    - **Fields**
      - **Columns** : Amount of columns that are displayed side by side in the block.
      - **Position** : The way that the fields are displayed in the block :
        - **Below** : The fields are displayed one below the other respecting the amount of columns.
        - **Beside** : The fields are displayed one beside the other respecting the amount of columns.
        - **Line** : The fields are displayed one beside the other without the tabulation.
    - **Organization**
      - **Next** : The way that the blocks are displayed in the page:
        - **Below** : Indicates that the next block will be placed below the current one.
        - **Beside** : Indicates that the next block will be placed beside the current one.
        - **Tabs** : Indicates that the next block will be placed in a different tab then the current one.
      - **Width** : Specifies the width that block will occupy in pixels or percentage, in case the value is in percentage, inform the (%).
      - **Collapse** : Enables the option to close the block.

- Create New Block

  To include new blocks in an Application, click on the button . Next, you’ll see the following interface to define the name and label of the block. At the end click on Create.

Creating application blocks configuration

- **Attributes**
  - **Name** : Name of the Block.
  - **Label** : Title of the block that'll be displayed in the application.

- Edit Blocks

  To edit a block just click on the icon , that is on the left side of the block. Next, you'll see the following interface to define the parameters of the blocks. At the end click on save.

- **Name** : Name of the block.
- **Title** : Block title for display.
- **Display Title** : This option, when active, allows to display the block title.
- **Title Font** : Font applied to the block title.
- **Font Size** : Size of the font applied to the block title.
- **Font Color** : Font color for the block title.
- **Background Color** : Background Color of the block title.
- **Background image** : Background image for the block title.
- **Title Height** : Height in pixels of the block title line.
- **Horizontal Alignment** : Horizontal Alignment of the block title (Left, Center and Right).
- **Vertical Alignment** : Vertical Alignment the block title (Top, Middle and Bottom).
- **Display Label** : Display the labels of the fields in the block.
- **Columns** : Amount of field columns in a block.
- **Columns Width** : How the width of the block is defined.
- **Label Color** : Color of the field labels.
- **Fields Organization** : How the fields are organized in the block.
- **Label Position** : Position of the field labels relating to the data.
- **Next Block** : Position of the next block relating to the current block.
- **Border Color** : Border color for the block.
- **Border Width** : Border Width for the block.
- **Block Width** : Width for the block.
- **Block Height** : Height for the block.
- **Cell Spacing** : Cell Spacing in the block.
- **Collapse** : Enables the option to close the block.

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos

Layout Settings

On this interface, you can define the theme for display of a specific application, this being because ScriptCase uses a Standard Definition of Values per project, that besides the Display Theme, allows to define values for other attributes on a Project level.

Header Template:

Allows to choose the template that’s going to be used as the Header of the application.

Footer Template:

Allows to choose the template that’s going to be used as the Footer of the application.

Button:

Allows to choose the button theme for the application.

Themes:

Choose one of the existing themes, it’ll load the display mode (colors ,fonts, etc.) that’ll be part of the application.

Header & Footer

Header

In this block, it’s the definition of the variables content that’ll be part of the header.

This page may change depending on the header format chosen in the Layout Settings.

Display Header:

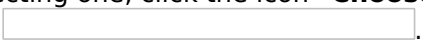
This option determines if the header will display.

Title:

Allows to inform the title displayed in the application.

Header Variables:

The field variables can be informed with anyone of displayed in the Combo box. Depending on the type, it’ll be necessary to associate the content with the field. Below there are the types of content:

- **Field** : When you choose the option “**Field**”, it’ll open a Combo box beside to choose the field you want. Selecting the desired field, it’ll associate the value of the field with the header.
- **Title** : This option when selected it’ll display in the header the value informed in the “**Application Title**”.
- **Date** : When selected the “**Date**” type, it’ll display the system’s date in the header. There are a diversity of formats using the date and time of the server. The format can be informed in the text field that appears beside the field. To access the existing formats, click on the icon and you’ll view a page display the formats.
- **Image** : When selecting the image type, it displays a field to inform the name of the existing image in the server. To locate the images existing and selecting one, click the icon “**Choose Image**” and to upload new images click on the button “**Upload**”  .
- **Value** : When selecting the type “**Value**”, the content informed in the text field that appears beside, it’ll be displayed in the header, you can inform texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

  Depending on the Application, you may have more than one title option.

Footer

This page may change depending on the footer format chosen in the Layout Settings.

Display Footer:

This option determines if the footer will display.

Footer Variables

The field variables can be informed with anyone of displayed in the Combo box. Depending on the type, it’ll be necessary to associate the content with the field. Below there are the types of content:

- **Field** : When you choose the option “**Field**”, it’ll open a Combo box beside to choose the field you want. Selecting the desired field, it’ll associate the value of the field with the footer.
- **Date** : When selected the “**Date**” type, it’ll display the system’s date in the footer. There are a diversity of formats using the date and time of the server. The format can be informed in the text field that appears beside the field. To access the existing formats, click on the icon and you’ll view a page display the formats.
- **Image** : When selecting the image type, it displays a field to inform the name of the existing image in the server. To locate the images existing and selecting one, click the icon “**Choose Image**” and to upload new images click on the button “**Upload**” .
- **Value** : When selecting the type “**Value**”, the content informed in the text field that appears beside,

it’ll be displayed in the footer, you can inform texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

Dynamic Search Settings

On this screen, the general behavior of the dynamic filter in the application will be defined.

Filter type

Defines the type of filter that will be used in the application.

Filter Builder

Allows the end user to create filters during the use of the application, combining different conditions and logical operators (AND/OR) to create more robust filters and generate more qualified reports.

Below is an example of how the filter builder works.

The user can add multiple fields or nested conditions to combine the use of logical operators.

Dynamic filter

Allows the end user to build filters during the use of the application by adding the fields that should compose the filter according to their needs, within the same logical operator.

Below is an example of how the dynamic filter works.

The user must select the logical operator that will be used for all filter fields: All conditions (AND) or any condition (OR).

Use the ENTER key for

This attribute defines the behavior of the ENTER key when the filter screen is being displayed.

Grid Filter Builder Settings

Enable/Disable

This attribute defines whether the user can disable fields or nestings in the filter builder, keeping them disabled until the application is reloaded.

Example of the disable button

Display filter condition

Defines whether the filter condition will be displayed while the user builds their filter.

When enabled, the condition is assembled as the user adds the filter’s conditions and logical operators. When disabled, this condition will not be displayed.

Modal Filter

Defines the location where the filter builder will open in the application.

When enabled, the filter will open in a modal. If disabled, the filter will be displayed in a div below the application’s toolbar. See below for examples of both behaviors.

Opening in a modal

Div below the toolbar

Highlight results

This attribute defines whether the search results will be highlighted in the application.

> The highlight color is defined by the application’s theme.

Example of highlighted results after search

Save Search in Filter Build (Grid)

This feature allows the end-user to save his searches in a profile. You can create some rules, like to save the searches by user login.

 Save Filter Interface.

Save Filter Interface.

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos

Select Fields for Filter Build in Grid

On this screen, the fields that will be available for use in the dynamic filter must be defined.

In the **left column**, all application fields and the **Search** field are displayed, which helps in locating the fields. In the **right column** are the fields selected to be displayed in the filter.

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

- - Move all fields to the right.
- - Moves only selected fields to the right
- - Moves only the selected fields to the left.
- - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the button.

> To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning

Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

> Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

Select Fields for Dynamic Search in Grid

On this screen, the fields that will be available for use in the dynamic filter must be defined.

In the **left column**, all application fields and the **Search** field are displayed, which helps in locating the fields. In the **right column** are the fields selected to be displayed in the filter.

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

- - Move all fields to the right.
- - Moves only selected fields to the right
- - Moves only the selected fields to the left.
- - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the button.

> To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning

Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

> Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

05 Quick Search

Quicksearch is an option that allows to search data in various fields of the application by using the text box in the toolbar.

Quick Search Settings

In the quick search settings you can find the following options:

Quicksearch layout

Simplified: Standard quicksearch view. This option displays the input where we must inform the terms to be searched for.

Extended: In this layout, in addition to the simplified layout options described above, it is possible to select the columns where a search will be carried out and the criteria to be selected.

The criteria must be selected in the option **Research Criteria**, they are:

Contains, Equal beginning, Exactly equal, Does not contain, Different, Greater than, Greater than equal, Less than and Less equal

Highlight results

Highlights the result of the research performed.

Maintain search condition

This option defines the behavior of quicksearch in relation to the advanced filter.

When activated, quicksearch takes into account the search performed in the advanced filter, so the quicksearch search will be carried out in conjunction with the advanced search.

When you disable the option, quicksearch completely disregards the search performed in the advanced filter.

Button within the search area

Defines the placement of the search button, whether it will be in the text area or outside it.

Option Enable -

Option Disabled -

Display combobox in the simplified layout

When active, display combobox with the fields selected in the **Quicksearch extended fields** option

Watermark

Watermark that will be displayed in quicksearch. A lang or a fixed text can be used.

By default, we use the lang: {lang_othr_qk_watermark}

Quicksearch width

Width in pixels of the Quicksearch input.

Settings for Individual Search by field

This configuration defines the fields that are available in the quicksearch combobox, when activating the Extended Layout option, in the general settings

Fields for individual search

Use fields displayed in the application: When selecting this option, only the fields displayed in the application will be available in the combobox.

Define fields manually: Allows the developer to select the fields that will be available in the quicksearch combobox.

Search settings in the ‘All Fields’ option

This option defines the fields that will be used in quicksearch, regardless of their configuration. That is, all fields selected in this option will always be available in the search.

Search in All Fields

All Fields added to the fields of the Individual Search : In addition to the fields selected in this option, the fields selected in the **Quicksearch extended fields** option will be available in the quicksearch search.

Only those selected for All Fields: Only the fields selected in this option will be available in the search through quicksearch

Search Criteria

Defines quicksearch search criteria.

When checking the options, the combobox with the criteria will be displayed

These are the options available as a quicksearch search criterion.

- **Contains**
- **Equal beginning**
- **Exactly equal**
- **Does not contain**
- **Different**
- **Greater than**
- **Greater than equal**
- **Less than**
- **Less equal**

Related Link

- Quicksearch

Related Video

- Quicksearch and Highlight option
- Macro sc_where_filter

Grid’s Search Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn’t insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid	✓	✓
Procedure	✓	✓
Chart	✓	✓
Form	✓	✓
Control	✓	✓
Calendar	✓	✓
Search	✓	✓
PDF Report	✓	✓

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven’t manage fields.

Text

General Settings

This type of field allows the developer to create quickly fields to display data from the database, where the final user can see the data in the way it was set by the developer.

Type of Data

Defines the type of the field in the application. In this case we should select **Text**.

Label

Defines the title that will be displayed in the field when the application is executed. The terminology used in the interface is fundamental to the system have a good usability, whe should use names and familiar terms to the final user of the application, instead of using terms from the system.

For example, this text field that has the name **customerid**, the client would have a much better understanding of the functionality of the field when we define the label as **Customer Name**.

Not only a fixed text, the **Label** attribute allows the use of langs to define the field's title, making it possible to use your application in a multi language project.

Case Settings

This option allows the developer to inform how the text will be displayed in runtime.

The options available are:

Lower Case: Every letter in the text will be converted to lower case.

Upper Case: Every letter of the text will be converted to upper case.

Capitalize first word: The first letter of the first world will be converted to upper case.

Capitalize all words: The first letter of every words will be converted to upper case.

Show HTML Content

When this option is active every HTML, CSS and JavaScript content that are in the database will be displayed with the main value.

Field Mask

Defines the field mask. There are two types of mask described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

For example, it is possible to set the mast to display a telephone number:

It will be show with this format on runtime:

It is also possible to set the field mask like those examples:

Field mask examples:

Telephone number

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Repeat Value:

When this option is active the value of the field will be repeated if the previous database register is the same.

Example:

SQL Type

Informs the type of the field in the database.

Field Behavior

Use autocomplete:

Field automatically turns into autocomplete according to existing values in the database.

Use Select2

Uses the new component for data selection, allowing searches within the select comboBox.

Width for the Select2

Sets a width for the area in the Select2.

Amount of characters

Sets the amount of characters to start the search.

Amount of rows

Sets the maximum number of rows to list the search result.

Width:

Defines a width in pixels for a result box.

Search options:

Defines the validation that will be made to fetch the search result.

Start equals to: Will return the records with the same start value as in the database.
Any part: Will return the records when exist the character in any part of the record.
End equals to: Will return the records with the same final value as in the database.

Position between values:

Defines the position that objects will be displayed.

Text Between Values

Text that will appear when using a filter condition between two values.

OnChange Submit:

Submit search on this field changing.

Show Condition:

To show or not the search condition. It only works if the search has at least one condition.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.

- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Dynamic Search
- Advanced search with relative periods
- Refine search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Integer

On this page, you will learn how you can configure settings related to the Number field. From the use of specific symbols display to the mode in which they are displayed. And thus, boost the application.

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Label:

It sets the title that will be displayed in the field when you run the application. The terminology used in the interface is fundamental for your system has a good usability. In this case, we recommend you to use names and terms familiar to the end user of the application, instead of using terms originated in the system.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Repeat value:

This option when enabled will allows you to repeat the field value if it is equal to the value of the previous record in the database.

Example:

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Color for the negative values:

It allows you to define a color when the value is negative, improving the understanding of the end user about that kind of value.

Example:

c

Display the value in words:

The value of the field will be displayed in full on application. This feature can facilitate the comprehension and understanding of the user.

Example:

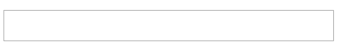
Line size:

Maximum size in characters to be displayed in the value cell, in full. When this value is exceeded the line will break within the cell.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:



Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set

in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

Related Videos ▷

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic search
- Advanced search with relative periods
- Refine search
- Search with blocks
- Pivot Table with Filter
- Highlight option
- Defined Search and Highlight option
- Quick Refined Advanced and Dynamic search
- Pivot Table with Filter

Decimal

General Settings

Decimal field Configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Field Behavior

Use autocomplete:

Field automatically turns into autocomplete according to existing values in the database.

Use Select2

Uses the new component for data selection, allowing searches within the select comboBox.

Width for the Select2

Sets a width for the area in the Select2.

Amount of characters

Sets the amount of characters to start the search.

Amount of rows

Sets the maximum number of rows to list the search result.

Width:

Defines a width in pixels for a result box.

Search options:

Defines the validation that will be made to fetch the search result.

Start equals to: Will return the records with the same start value as in the database.
Any part: Will return the records when exist the character in any part of the record.
End equals to: Will return the records with the same final value as in the database.

Position between values:

Defines the position that objects will be displayed.

Text Between Values

Text that will appear when using a filter condition between two values.

OnChange Submit:

Submit search on this field changing.

Show Condition:

To show or not the search condition. It only works if the search has at least one condition.

Values format

Decimal Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator, Negative sign and Negative number format.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Advanced search with relative periods
- Refined Search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Currency

General Settings

Currency field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Currency, you can currency values to the field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Currency Field Behavior Interface of the Search Configuration.

- **Use auto-complete** : The field behaves as an auto-complete according to the values existing in the database.
- **Amount of characters** : Sets the amount of characters to start the search.
- **Amount of rows** : Sets the maximum number of rows to list the search result.
- **Width** : Sets the width in pixels for the result box.
- **Search options** : Defines the validation that will be made to fetch the search result.
- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Values format

Currency Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).

    - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the title of the field.
    - **Height** : To define a height for the title of the field.
- **CSS of the Field**
    - **Font** : Allows to choose the font type, that will be applied to the application field.
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the field.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
    - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the field.
    - **Height** : To define a height for the field.
- **CSS of the Input Object**
    - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the Input Object.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
    - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the Input Object.
    - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
    - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

    - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

    - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search

Related Videos

- Advanced search with highlighting
- Define Refined Search
- Search with Filter
- Pivot Table with Filter
- Advanced search with date periods
- Refined Advanced and Dynamic search
- Search with Slicers
- Pivot Table with Filter

Date

General Settings

- **Data Type** : Define the type of field for the application. When set to Date, you can inform a date.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Watermark**: Displays a watermark in the field input.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the datatype of field in the database.

Values format

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator attribute.
- **Date separator** : Allows you to inform the separator symbol for the date.
- **Display** : Select the format of the day for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **Date**. You need to use the characters **A**, **M** and **D** that correspond to **Year**, **Month** and **Day**.
- **Use Combo-box** : Allows you to select the date using a combo-box.
  - **Year as Combo** : Allows to use the year combo to select the date.
    - **Initial Year** : First year displayed in the combo.
    - **Actual Year +** : Display the current plus the amount of years informed.
- **Month in full textual** : Displays the Month format in Full.
- **Display Calendar** : Enables the a calendar icon beside the field, this allows to select the date from a calendar with the format already setup.
  - **New Calendar** : Defines if the JQuery calendar (New Calendar) is going to be displayed or the old format.
  - **Years Limit** : Amount of years displayed in the calendar.
  - **View week number** : Displays the number of the week in the application.
  - **Additional months** : Displays the additional months of the calendar.
  - **Show Combo year and month** : Displays the year and month of the calendar in the combo box.

Field Behavior

- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Special Conditions

- **All Period** : Searches for all periods of dates.
- **Today** : Searches in today’s date.
- **Yesterday** : Searches in yesterday’s date.
- **Last 7 days** : Searches the last 7 days. Ex: ((01/01/2017 01/07/2017).
- **This month** : Searches the dates from the first day of the current month.
- **Last month** : Searches the dates from the first day of lasts month.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of

Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refine Search
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Time

General Settings

Time field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Time, you can inform a time to this field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Values format

Time Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator and time separator attributes.
- **Time separator** : Allows you to inform the separator symbol for the time.
- **Display** : Select the format of the time for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **TIME**. You need to use the characters **HH**, **II**, and **SS** that correspond to **Day**, **Hour**, **Minutes** and **Seconds**.

Field Behavior

Time Field Behavior Interface of the Search Configuration.

- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.

- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Search with Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos

Datetime

General Settings

Datetime field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Datetime, you can inform a date and time to this field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Values format

Datetime Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator and time separator attributes.
- **Date separator** : Allows you to inform the separator symbol for the date.
- **Time separator** : Allows you to inform the separator symbol for the time.
- **Display** : Select the format of the day/time for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **DATETIME**. You need to use the characters **A**, **M**, **D** , **HH**, **II**, and **SS** that correspond to **Year**, **Month**, **Day**, **Hour**, **Minutes** and **Seconds**.
- **Use Combo-box** : Allows you to select the date using a combo-box.
  - **Year as Combo** : Allows to use the year combo to select the date.
    - **Initial Year** : First year displayed in the combo.
    - **Actual Year +** : Display the current plus the amount of years informed.
- **Month in full textual** : Displays the Month format in Full.
- **Display Calendar** : Enables the a calendar icon beside the field, this allows to select the date from a calendar with the format already setup.
  - **New Calendar** : Defines if the JQuery calendar (New Calendar) is going to be displayed or the old format.
  - **Years Limit** : Amount of years displayed in the calendar.
  - **View week number** : Displays the number of the week in the application.
  - **Additional months** : Displays the additional months of the calendar.
  - **Show Combo year and month** : Displays the year and month of the calendar in the combo box.

Field Behavior

Date Field Behavior Interface of the Search Configuration.

- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Special Conditions

Datetime Field Special Conditions.

- **All Period** : Searches for all periods of dates.
- **Today** : Searches in todays date.
- **Yesterday** : Searches in yesterdays date.
- **Last 7 days** : Searches the last 7 days. Ex: ((01/01/2017 01/07/2017).
- **This month** : Searches the dates from the first day of the current month.
- **Last month** : Searches the dates from the first day of lasts month.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**

- **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links 🔗

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos ▷

Select

General Settings

Select field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Select, you can select multiple option from a combo box (Select Field).
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Select Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

  Selecting the lookup type.

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimitation.

      - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Height** : Defines the height for the select object.
    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to

a value informed manually (Title internal value).

- **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
- **Title** : Text displayed on the title of the Select object.

- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.
- **Link** : Allows to create a link to another form allowing to manipulate the list displayed on the select field. After the manipulation, the select object it updated automatically.

- **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Height** : Defines the height for the select object.
  - **Multiple Values (delimiter)**

    You can store various values for the select field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Height** : Defines the height for the select object.
  - **Multiple Values (position)**

    Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of

bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Height** : Defines the height for the select object.

- **Multiple Values (binary)**

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

Editing Lookup Configuration Multiple Values (Binary)

- **Label** : Text that will be displayed in the item list of the select field.
- **Height** : Defines the height for the select object.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  Displaying the original lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Lookup Method - Actual value

This lookup is used to list all the values in the selected field.

This lookup will apply a “distinct” to your SQL query.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Dynamic Search
- Advanced search with relative periods
- Refined Search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Double Select

General Settings

Double Select field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Double Select, your allowed to have multiple options selected.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Double Select Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, getting these values from the database.

Lookup Settings Display for the field. *Automatic Lookup Interface..*

- **SQL Select Statement** : Defines the SQL command that will get the values displayed on the form field. The command needs to have the following structure:

  ```
  SELECT key_field , display_field FROM table
  ```

  The value of the key_field will be stored in the table field.

- **Height** : Set the height(lines) of the field interface.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter than (;).
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**

- **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links 🔗

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos ▷

Check box

General Settings

Check box field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Check box, your allowed to have multiple options selected.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Check box Field Behavior Interface of the Search Configuration.

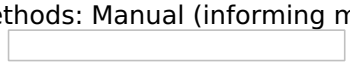
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).  *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the CheckBox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
    - **Option check all** : Displays two options on the field to check and uncheck all.

    - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Columns** : Set the amount of columns, for the list of items.
  - **Multiple Values (delimiter)**

    You can store various values for the checkBox field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Columns** : Set amount of columns, for the list of items.
  - **Multiple Values (position)**

    Stores a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

    For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

    The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

    Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the

following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Columns** : Set the amount of columns, for the list of items.

- **Multiple Values (binary)**

  Stores a diversity of information selected on the field in only one value.

  In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  To store the data, it is done an internal sum of the decimal number. See the examples below.

  Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 2 (Culture) = **3**

  Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

  Setting up Multiple Values (Binary)

  - **Label** : Text that will be displayed in the item list of the checkBox.
  - **Columns** : Allows you to inform the amount of columns, for the list of items.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Saves all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Refreshes the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Option check all** : Displays two options on the field to check and uncheck all.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio,

Select, Text, etc
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

    Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

    Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

    Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos

Radio

General Settings

Radio field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Radio, your allowed to select one of the options listed.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Radio Field Behavior Interface of the Search Configuration.

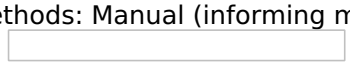
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).  *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Columns** : Allows you to inform the amount of columns, for the list of items.

    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

      - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
      - **Title** : Text displayed on the title of the Select object.

- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

  This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

  Manual Lookup Interface.

  - **Label** : Text that will be displayed in the item list of the radio.
  - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
  - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
  - **Insert Button** : Adds to the list the values informed on the fields Label and Value.
  - **Update Button** : Modifies the attributes of the selected item.
  - **Remove Button** : Remove the selected item from the list.
  - **Clear Button** : Clear the Fields.
  - **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on load lookup definition.
  - **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
  - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

    - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
    - **Title** : Text displayed on the title of the Select object.
  - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
  - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

    Displaying the original lookup.
  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Defined Search
- Quicksearch and Highlight option
- Search Refined by Advanced and Dynamic search
- Pivot Table with Filter
- Macro se_where_filter
- Highlight option

Related Videos

Text Auto-Complete

General Settings

Text Auto-Complete field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Text auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal Text for the data.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Text Auto-Complete Field Behavior Interface of the Search Configuration.

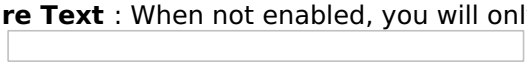
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **Case Settings** : Allows to convert the letters of the field when losing the focus. The options are:
  - **Upper Case** : All in Upper Case.
  - **Lower Case** : All in Lower Case.
  - **Capitalize first word** : Capitalizes the first letter of the word.
  - **Capitalize all words** : Capitalizes the first letter of all the words.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below.  *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Defined Search
- Quicksearch and Highlight option
- Search Refined by Advanced and Dynamic search
- Pivot Table with Filter
- Macro se_where_filter
- Highlight option

Related Videos

Number Auto-Complete

General Settings

![Number Auto-Complete Field Behavior Interface of the Search Configuration.][filtro_cons_número_auto]
Number Auto-Complete Field Behavior Interface of the Search Configuration.

- **Data Type** : Define the type of field for the application. When set to Number auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal number for the data.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Number Auto-Complete Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Values format

Number Auto-Complete Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator, Negative sign and Negative number format.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below. *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps

the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value

Related Videos

- Dynamic Search
- Advanced search with relative periods
- Quick search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Date Comparison

Special field that allows you to change the functioning of a date type field in the summary filter, enabling the comparison of metrics over time. To use this type of field, it is necessary to have a date type field created, and later change its type to Date Comparison.

> It is only available in the advanced filter and its operation is exclusive to the query summary module.

Example of the field in the summary filter

Example of the Generated summary

When using the comparison, the summary will be assembled showing the values for the periods informed.

General Configuration

In this block are the general settings of the field, and where it will be 'transformed' into a comparative data field, simply access the settings of a data field and change the *Data* type to **Date comparison**

Data type of the field.

This attribute defines the type of field in the application. This definition implies the configuration options that will be displayed to the developer as well as the behavior of the field when running the application.

During the application creation process, Scriptcase maps the fields of the table(s) used and defines the data type of the fields according to their SQL type.

For example
Fields of type varchar in the table will be defined as text in the **Data Type** attribute.

> It is possible to change the field's data type, however it is important to take into account the **SQLType**, displayed in the General Configuration block, to define the field's data type.

Search Label

Defines the title of the column displayed in the executed application and can be defined using: Lang, Text or Global variable.

Lang

The use of lang in the attribute definition is recommended for creating multilingual systems. The langs must be previously defined using the option Traduzir aplicações or through Dicionário de Dados.

Fixed text

In this case, the column title will not be changed due to user interaction, such as changing the application language, for example.

It is a good option for systems created with just one language.

Global variable

The use of global variables allows you to change the column title according to your system business rule.

> We also provide the macro sc_label which allows you to change the column title regardless of how the label attribute was defined.

Example of a column title

Use same label used on the Grid

This option, when activated, allows you to use the title of the field defined in the grid, in this case the value defined in the **Label filter** attribute will be disregarded.

Value Formatting

Regional Settings

Allows you to apply regional date formatting settings to the field, according to the application language. When disabling the use of *regional settings*, some settings will be displayed such as: **Date separator**, **First Day** and **Display**.

Check below for more details about each configuration option.

> It is possible to change regional settings in general, allowing the use of the option and the same configuration in all your projects, according to the language used. For this configuration, access the Menu *Locales* > *Configurações Regionais*

Date separator

Defines the character that will be used to visually separate the parts of the date (day, month and year). The characters normally used are: slashes (“/”), hyphens (“-“) and periods (.).

Alguns exemplos

- **slashes (/)**: 25/10/2023
- **hyphens (-)**: 25-10-2023
- **periods (.)**: 25.10.2023

> This option will be displayed if the **Use regional settings** attribute is disabled.

First day

Defines the first day of the week to be displayed in the calendar in the date field, for selecting dates.

First day change example

In this example, the *First day* was defined as Monday, changing the order in which the days of the week are displayed in the field calendar.

> This option will be displayed if the **Use regional settings** attribute is disabled.

Display

Defines the date display format by selecting one of the listed formats.

The options are:

- dd-mm-aaaa
- mm-dd-aaaa
- aaaa-mm-dd

> This option will be displayed if the **Use regional settings** attribute is disabled.

View week number

Defines whether the number of weeks will be displayed in the date field calendar.

Example of a calendar with number of weeks

Special Conditions

Defines the special conditions that will be displayed in the date field calendar, see the example below.

Example of special conditions in the date field calendar

If no interval is enabled, the calendar in the date field will not have any special conditions.

Example of the date field without special conditions

Select Values

In this attribute, the special conditions that will be listed as an option in the application must be defined. The predefined intervals are grouped according to the period they refer to. For example, the condition *Last week* is in the tab **Week**.

In each of the tabs it is also possible to create personalized intervals using the button Add new range.

In the tab , all selected ranges will be listed, so that the display order can be changed according to the user's needs.

Sorting tab

In this tab, all selected conditions will be listed, allowing the intervals to be reordered according to the system's needs. You can also remove conditions one by one or using the button **Remove all**.

Add New Range

Clicking on *add new range* will open a screen with some options for adding new ranges. To add an interval, we must select the type (next or last) and enter the period (value).

For example, adding a range in the year tab, defined *Last* in the attribute as **Type** and reporting *2* in the attribute **Value** a range will be created: **Last two years**

> This button is available on the **Year**, **Quarter**, **Month**, **Week** and **Day** tabs.

Type

Defines whether the created interval will refer to a future period (next) or a previous period (last).

- **Next** - Defines that the created interval will refer to a future date, for example, **Next year**.
- **Last** - Defines that the range will refer to a previous date, for example, **Last Year**.

Value

Sets the amount of time for the interval.

> This field only accepts positive integer values greater than zero.

Title

Defines the text that will be displayed in the application. When creating a new range, a lang is automatically

generated, it is also possible to use a fixed text.

Include Current

This option changes how the created interval works, adding the current period to the interval.

For example, adding an interval to the year tab, defining *Next* in the **Type** attribute and entering *2* in the **Valor** attribute and checking the **Include current** option will create an interval: **Next 2 years from current**.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse

leaves the help icon.

- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

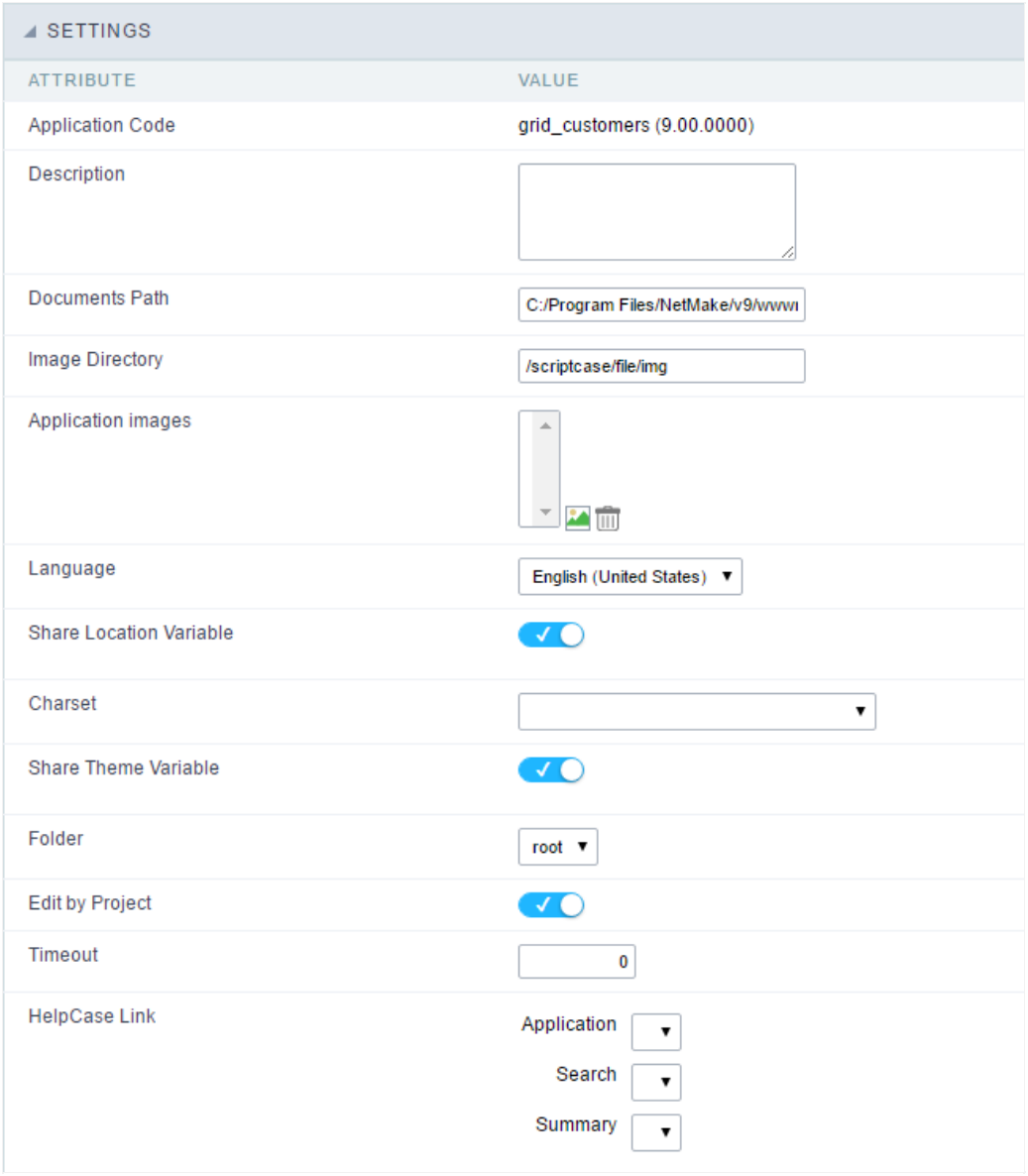
Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Application Settings (Grid)

Settings

With this interface, you can set the common attributes of the app.

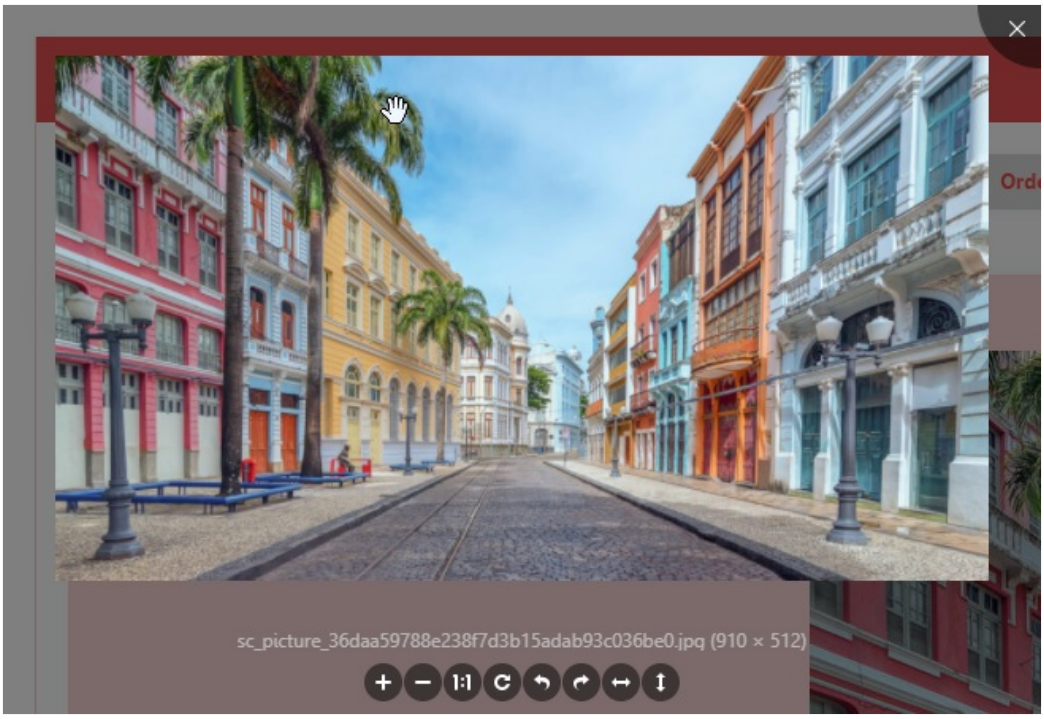


Application Settings Interface

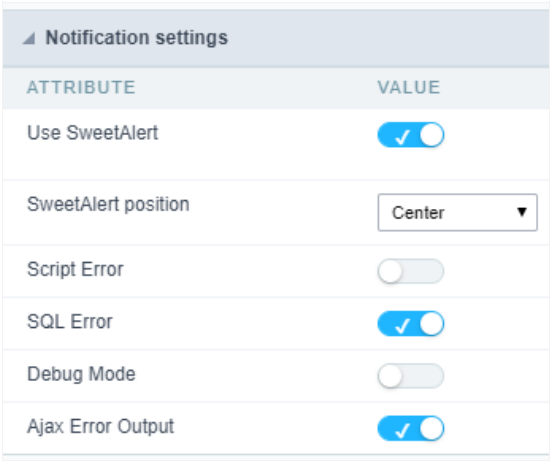
- **Attributes**
  - **Application Code** : It is the name that defines an application. An app can be renamed at the List of Application.
  - **Description** : This field contains a brief description of the application objectives.
  - **Documents Path** : The absolute path to store uploaded documents in the application.
  - **Image Directory** : The filesystem directory to store the application images.
  - **Application images** : Import images into the application to allows using them in the application.
  - **Language** : Set the default language of the application. Display all the application hints and messages in the selected language.
  - **Share Location Variable** : Define if the app shares the regional settings with other applications through a session variable.
  - **Charset** : Define a specific charset to use in the application.
  - **Share Theme Variable** : Define if the app shares the Theme settings with other applications through a session variable.
  - **Folder** : Define the project folder that contains the app.

- **Edit by Project** : Define if other project developers can edit the application.
- **Timeout** : Set the session runtime timeout in seconds. If the value is Zero, it assumes the default timeout of the PHP.
- **HelpCase Link** : It allows to associate a HelpCase file with the application.

- **Image Viewer**: Activates the image viewer features in the running application. By clicking on the image the user can drag, rotate, invert, and enlarge the selected image.



Notification Settings

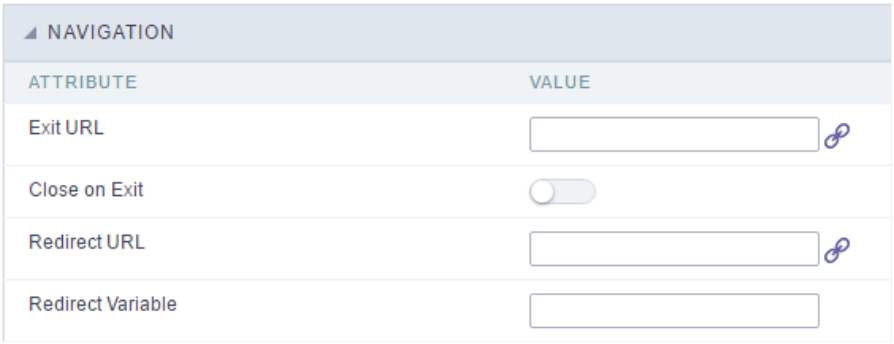


- **Use SweetAlert**: Use the SweetAlert to display messages from the application. When this option is active, it will replace the browser's "confirm" and "alert".
- **SweetAlert position using Toast** : The position to display error messages on the application.

- **Script Error** : Allows displaying the line code where there is an error..
- **SQL Error** : Allows displaying the SQL statement if it got an error.
- **Debug Mode** : Runs the application in Debug mode, showing all SQL statements the application is executing.
- **Ajax Error Output** : Enables the Ajax alert for debugging errors.

Navigation

This interface allows defining the navigating behavior of the application



Navigation Interface.

Exit URL

URL to where the user goes when he clicks on the “exit” button.

Close on Exit

Close the browser window when the user clicks on the “exit” button.

Redirect URL

Redirect to another URL in case there aren’t any global variables available.

Redirect Variable

Creates a variable with the application name and sends it to the redirected application.

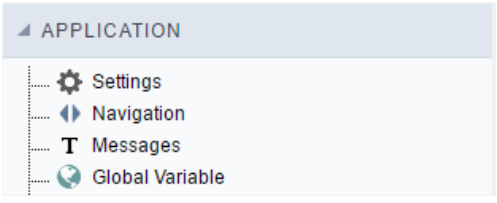
Global Variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**“**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target=”_blank”} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



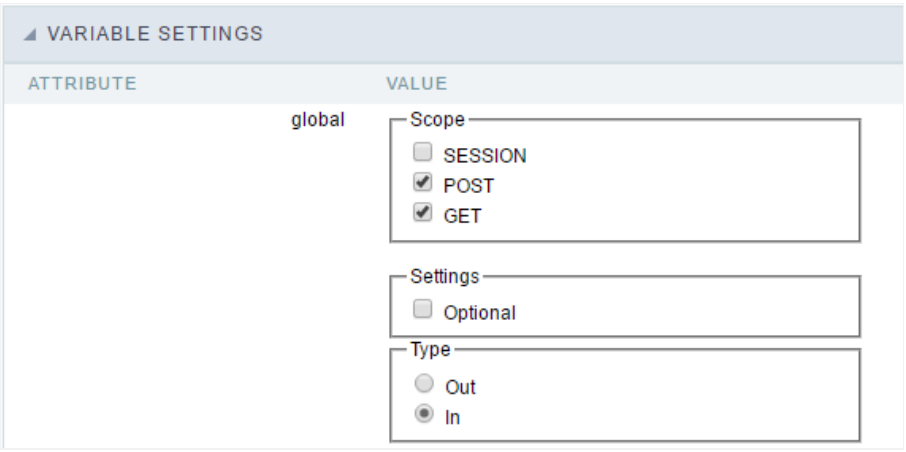
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

Indicates where the application is using the variable.

Grid’s Links Overview

This feature allows the developer to create links between applications of the same project, expanding the integration of applications. All link options are grouped under the Application Links menu.

Application Links

In the first access to the menu, we can see the list of existing connections in the application.

> If the application does not have a configured link, the application list screen will be displayed with the message: **This application does not have any link. Click here to create one now.**

ID

Link identification ID.

Type

The type of link created, some links such as Edit Link allow only one link per application. In this case, the developer will be able to check the connection types that already exist in the application

Target Application

This info show the target application name.

Actions

This column has edit options for the links.

Properties

It allows accessing the binding properties where it is possible to configure the binding behavior.

Link

Displays the links screen, where it is possible to configure the link that was made with the application informed in the target application column. In this option it is possible to change the parameter passed in the connection as well as the target application.

Delete

Permanently deletes the connection in the applications.

> Through the option Restore Applications, it is possible to get a previous version of the application, making it possible to recover the link deleted.

Links Type

The Grid application has the following links options

- Edit Link - Allows the creation of a link with form applications in order to edit the records listed in the grid.
- Capture Link: It allows the creation of links from the filter fields of the Grid application, in order to enable the recovery of the value to fill in the field, with another Grid application of the project.
- Field Link: Allows you to create a link through the application fields to any application in the project.
- Button Link: It allows creating a link through the buttons created by the developer in the query application, to any application in the project.

Grid Applications Link Types x Application available to link

Check the list of applications that can be associated with the available connection types.

The restrictions on the choice of applications are the **Edit Link** which only allows linking with a form and the **Capture Link** where it is possible to link the filter fields with only one other grid application.

	Edit Link	Field Link	Capture Link	Button Link
Grid				
Chart				
Form				
Control				
Search				
Menu				
Tree Menu				
Tabs				
PDF Report				
Dashboard				
Blank				
Calendar				

Related Link

- Drill Down
- Master/Detail on 3 levels
- Report link to edit the record
- Modal Links
- Macro sc_link

Related Video

Edit Link

Creating an Editing Link

This type of link allows the developer to create a link from a grid to a form with the objective of editing the register of a grid's row.

In the link options, we will choose the **Application Link**. When we choose this option, edit a register from a Grid will be possible.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Open in an iframe:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a iframe in one of the four options available in the iframe settings.

Open in a parent:

When we use this option the target application will be displayed in the same window of our application, and the target application will have a back button so we can return to the previous application.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window.

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set these following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Display the button new in the grid:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Also, when we select the option **Yes**, it is displayed two new options **Label to the New button** and **Hint to the New button**:

Label to the New button:

In this option we will inform a name for the button that will be displayed in the Grid.

Hint to the New button:

In this option we will inform a message that will be displayed when the mouse cursor is over the **New** button.

Shortcut key to the New button:

Indicates the shortcut key to the button add new register.

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Selecting this option will direct the application to the exit application or to the application that initially called the Form, after the record has been updated.

Close the form after an insert:

Selecting this option will direct the application to the exit application or to the application that initially called the Form, after the record has been inserted.

Open the application in a tab inside the menu:

Choosing the option **Yes**, the application will be open in a **menu tab**, when executed from a menu application, instead of opened in another browser window.

Display the button New in the grid:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Shortcut key to the New button:

Indicates the shortcut key to the button add New Register.

Label to the New button:

In this option we will inform a name for the button that will be displayed in the Grid.

Hint to the New button:

In this option we will inform a message that will be displayed when the mouse cursor is over the **New** button.

Open in an iframe.

When we use this option the target application will be displayed in the same window of our application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Shortcut key to the New button:

Indicates the shortcut key to the button add New Register.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Iframe properties

In this settings, we can define some iframe details that will display the target application.

Display the target application’s header:

When the option **Yes** is selected, the header of the target application is also displayed in the iframe.

Iframe position in relation to the main application:

In this option, we will define in which position in relation to the main application the iframe will be displayed, there are four options:

Below: The iframe will be displayed below the main application.
Above: The iframe will be displayed above the main application.
Right: The iframe will be displayed on the right of the main application.
Left: The iframe will be displayed on the left the main application.

Action after an insert:

In this option, we will define what will happen after a register insert, there are two options:

Reload the grid: The current page will be refreshed after the insert.
Move to the end of the grid : After the insert will be displayed the last page of the grid will the last register inserted.

Iframe height:

Allows to set the iframe height.

Iframe width:

Allows to set the iframe width.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Selecting this option will direct the application to the exit application or to the application that initially called the Form, after the record has been updated.

Close the form after an insert:

Selecting this option will direct the application to the exit application or to the application that initially called the Form, after the record has been inserted.

Display the button New in the grid:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Shortcut key to the New button:

Indicates the shortcut key to the button add New Register.

Label to the New button:

In this option we will inform a name for the button that will be displayed in the Grid.

Hint to the New button:

In this option we will inform a message that will be displayed when the mouse cursor is over the **New** button.

Form Properties

In those settings, we can define which buttons will be available in the target form application. Initially we have five options, that are:

Enable Insert button:

In this option we can define if the **New** button will be available in the target application.

Enable Update button:

In this option we can define if the **Save** button will be available in the target application.

Enable Delete button:

In this option we can define if the **Delete** button will be available in the target application.

Enable Navigation buttons:

In this option we can define if the buttons **first, previous, next and last** will be available in the target application.

Enable register editing button in the grid:

In this option we can define if the edit register button, which is the **pencil** in the grid, will be available.

Related Link

- Report link to edit the record

Related Video

- Linking Applications

Grid Field Link Settings

Creating a field link

Allows to create a link, represented through a link, from a grid's column to any project's existing application. All the links are displayed in a dropdown if there are more than one link to the same field.

In the types of link option, we will choose the **Field link**.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open the application in a tab inside the menu:

Choosing the option **Yes**, the application will be open in a **menu tab**, when executed from a menu application, instead of opened in another browser window.

Open in an iframe

When we use this option the target application will be displayed in the same window of our application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Iframe properties

In this settings, we can define some iframe details that will display the target application.

Iframe position in relation to the main application:

In this option, we will define in which position in relation to the main application the iframe will be displayed, there are four options:

Below: The iframe will be displayed below the main application.
Above: The iframe will be displayed above the main application.
Right: The iframe will be displayed on the right of the main application.
Left: The iframe will be displayed on the left the main application.

Iframe height:

Allows to set the iframe height.

Iframe width:

Allows to set the iframe width.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Grid’s Properties

In those settings we can define how our target grid will be displayed. We inicially have six options, that are:

Initial module:

We can define how our application will be executed in those two options:

Grid: The application will be executed as a grid.
Search: The application will be executed as a search.

Quantity of rows:

In this option we can define the quantity of rows that will be displayed in the target application.

Quantity of Columns:

In this option we can define the quantity of columns that will be displayed in the target application.

Pagination:

In this option we can define if the target application’s pagination will be total or partial, in case partial is the selected option the quantity of rows will be the informed previously.

Enable header:

In this option we can define if the target application’s header will be displayed.

Enable Navigation buttons:

In this option we can define if the buttons **first, previous, next and last** will be available in the target application.

Related Link

- Drill Down
- Master/Detail on 3 levels

Related Video

- Linking Applications
- Macro sc_link

Capture Link

Creating a Capture Link

The capture link is used to return a value from a **Grid** to a **Form** field.

In the types of links options, we will choose the **Capture Link**. After selection this option, we should also choose which field we want to return the value of.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Application with no parameters

However, when the target application does not have any defined parameters, the following screen is displayed:

Clicking in the button, you will be taken to the target application to create a parameter, so you can use the update button in the **parameters definitions** to refresh them.

Link properties

In this screen we will set the application display mode that will be called in the link.

In this type of link there is only one display option:

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Modal:

In this option we are going to define if the target application will be opened in a Modal.

Yes: This option will make the target application be opened in a modal. **No**: This option will make the target application be opened in a new window.

If **Yes** is selected in the previous option, the Modal **Height** and **Width** will be available.

![Modal with yes][modal2]

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

If **No** is selected in the previous option, only those options will be available.

Allows to modify manually in the update:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Allows to modify manually in the insert:

Indicates which shortcut key to the button add new register.

Allows to modify automatically in the update:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Allows to modify automatically in the insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Related Video ▷

- Linking Applications

Button Link

Creating a Button Link

Allows the developer to create a link where the call to the other application will be done through a button.

In the type of links options, we will choose the **Button Link**. Choosing this option it will be possible to create a link to any other application.

List of applications

After selecting this option, The list of applications to what you want to create a link will be displayed.

This screen can be viewed in the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open the application in a tab inside the menu:

Choosing the option **Yes**, the application will be open in a **menu tab**, when executed from a menu application, instead of opened in another browser window.

Choosing the option **Yes**, there are also some other settings to be displayed:

Title to the tab:

This option allows the developer to set a title to the tab that will be opened when used in a Menu application.

Hint to the tab:

This option allows the developer to set a message to be displayed when the mouse cursor is over the tab Menu.

Active tab icon:

This option allows the developer to set an icon to be displayed in the tab when used in a menu application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Height:

Allows to set the Modal's height.

Width:

Allows to set the Modal's width.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Form Properties

In those settings, we can define which buttons will be available in the target form application. Initially we have five options, that are:

Enable Insert button:

In this option we can define if the **New** button will be available in the target application.

Enable Update button:

In this option we can define if the **Save** button will be available in the target application.

Enable Delete button:

In this option we can define if the **Delete** button will be available in the target application.

Enable Navigation buttons:

In this option we can define if the buttons **first, previous, next and last** will be available in the target application.

Enable register editing button in the grid:

In this option we can define if the edit register button, which is the **pencil** in the grid, will be available.

Enabling the option **Enable navigation buttons**, one more option will be displayed:

Keep the WHERE clause in the target application:

Enabling this option, the WHERE clause of the grid application will be kept to the target application.

Related Video ▷

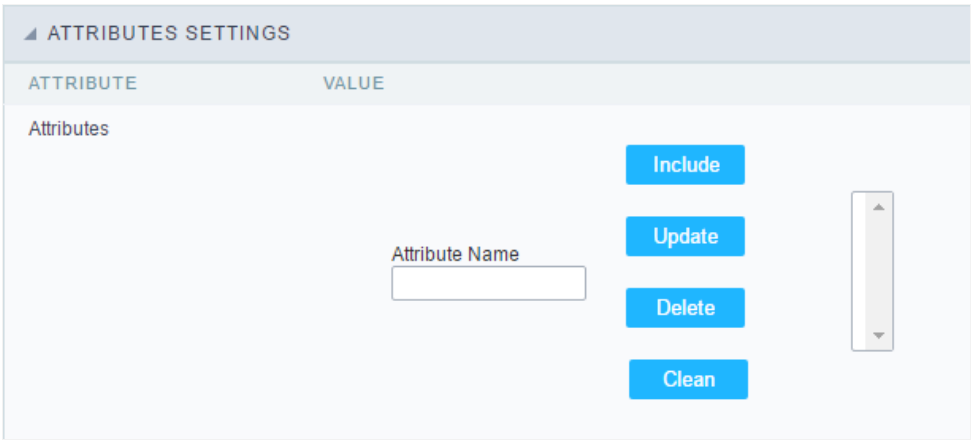
- Linking Applications

Grid Programming

In this version of ScriptCase is incorporated with the concept of programming with the use of attributes, methods, resources and libraries. In the previous versions it was already possible to create business rules in applications using this concept, but the big difference now is that this can be done in a more organized and dynamic way, facilitating both the development as the understanding of the rule by another developer.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal Libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

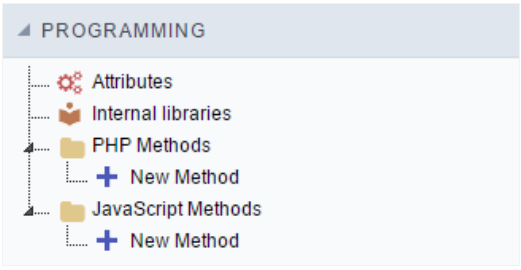
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

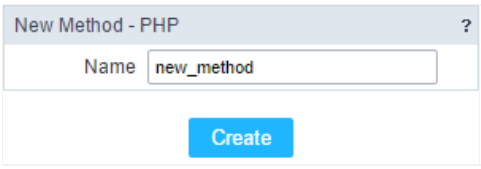
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

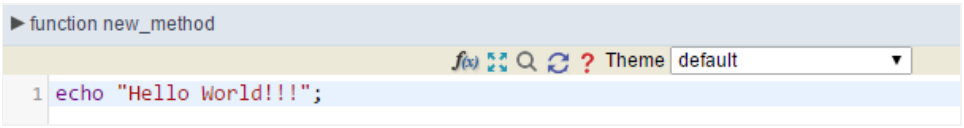


- Give the method a name and click Create. As in the image below.

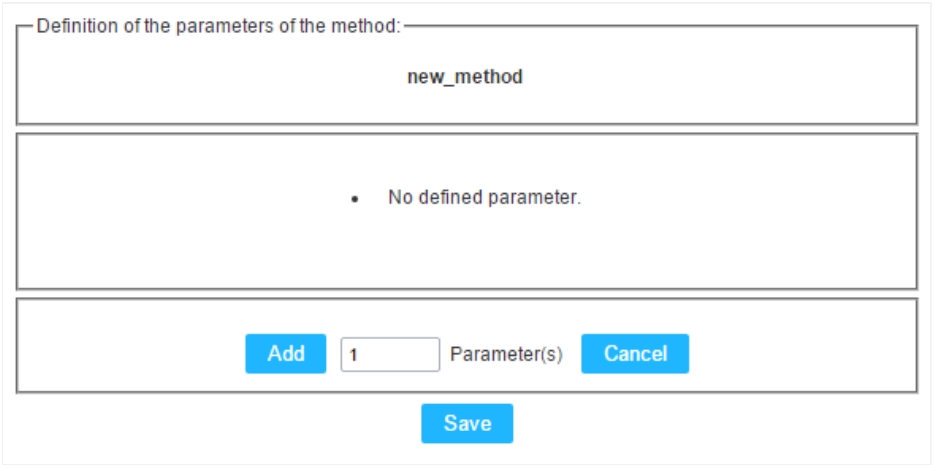
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



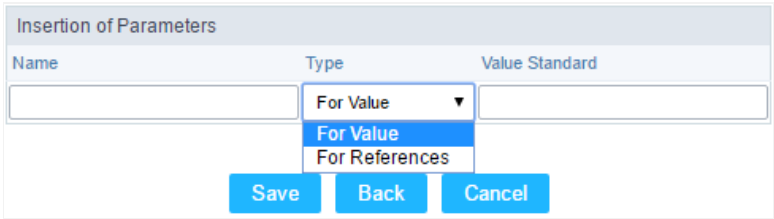
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

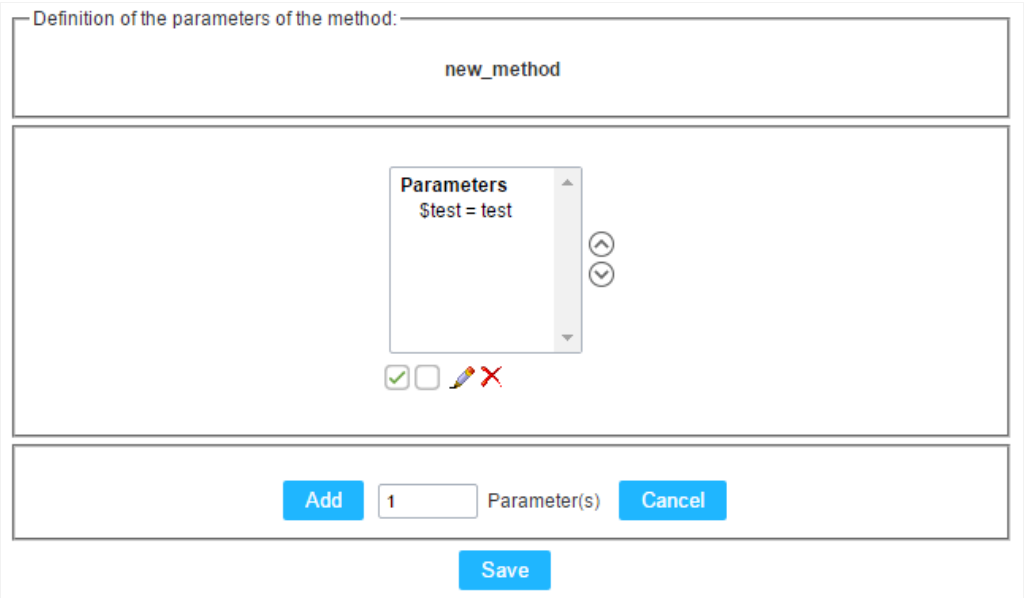


- **Name** : Type in the variable's name.

- **Type** : Selecting the type of variables: For Value or For Reference.
- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:



- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

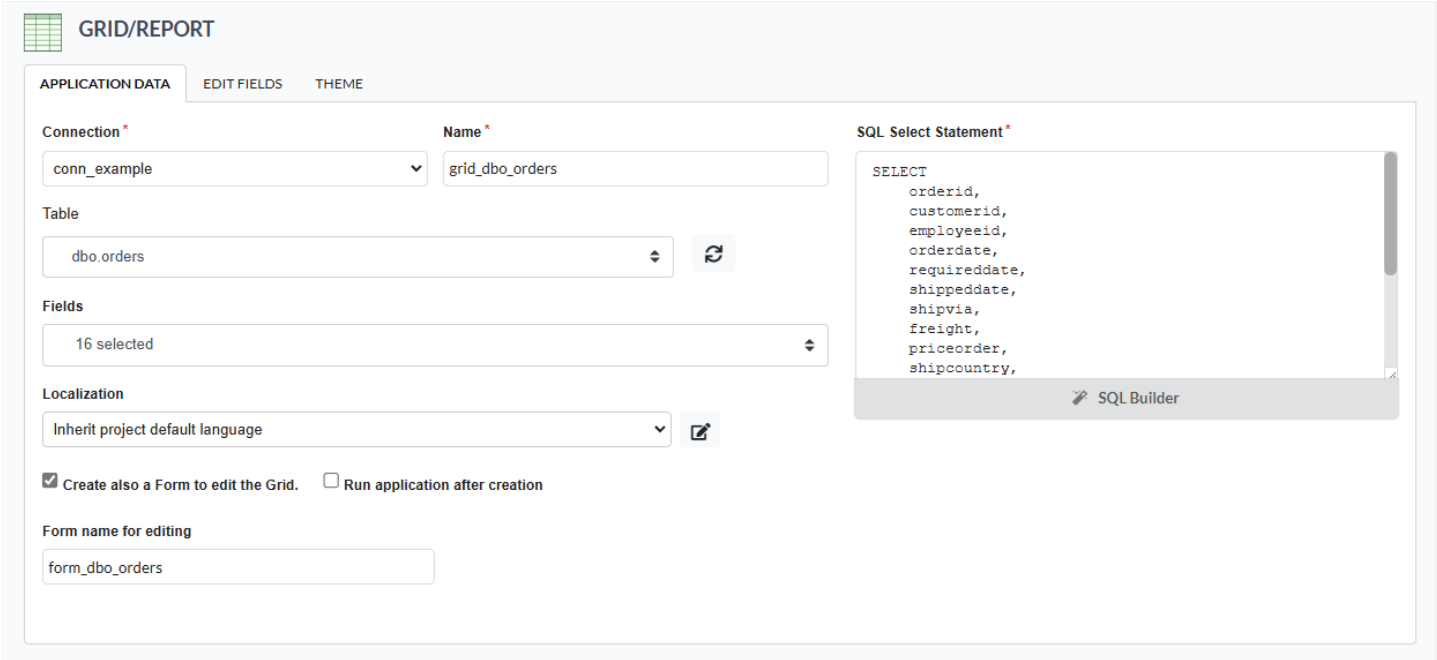
Creating a Grid/Report Application

New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.

Application Data



Connection

Defines the connection used for the creation of the application. The existing connections in the project will be listed.

Name

The name of the application being created. It cannot contain special characters.

Table

Defines the tables to be used in the application. (Form and Calendar can only use one table).

Fields

Defines the fields that will be part of the applications.

Localization

The language of the application to be created. The project's default language is automatically selected.

SQL

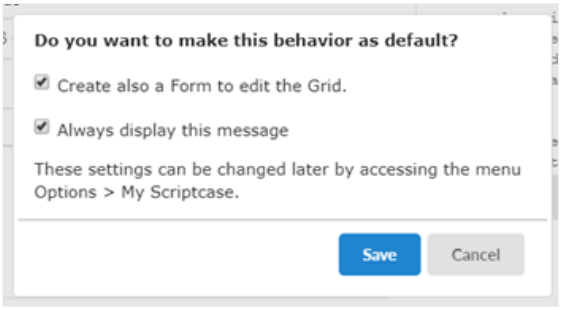
Displays the SELECT command generated after selecting the tables and fields. This field also allows the insertion of a previously created SQL statement, as long as it uses tables that exist in the database selected in the connection.

Create also a Form to edit the Grid

This option creates a form application linked to the grid, automatically establishing an edit link between the applications. For more details about the edit link, click here.

This option is available only when the grid is created using a table.

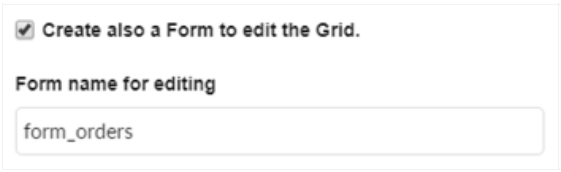
When this option is selected, a dialog box will appear asking the developer to define the default behavior for creating future grid applications.



- **Create also a Form to edit the Grid** - Defines the default behavior for creating future grid applications.
- **Always display this message** - Determines whether this dialog box will always be displayed.

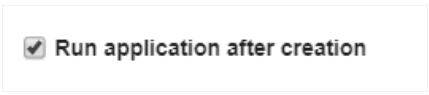
This configuration can be modified later by accessing the menu **Options > My Scriptcase**.

After confirming the default behavior, you will need to specify the name of the form to be created.

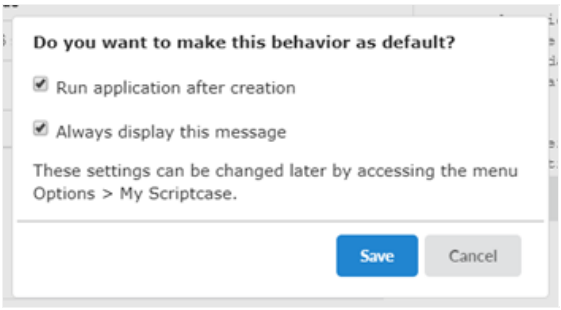


Run application after creation

This option defines whether the grid will be executed after it is created.



When this option is selected, a dialog box will appear where the default behavior of Scriptcase for creating other grids in the project must be defined.



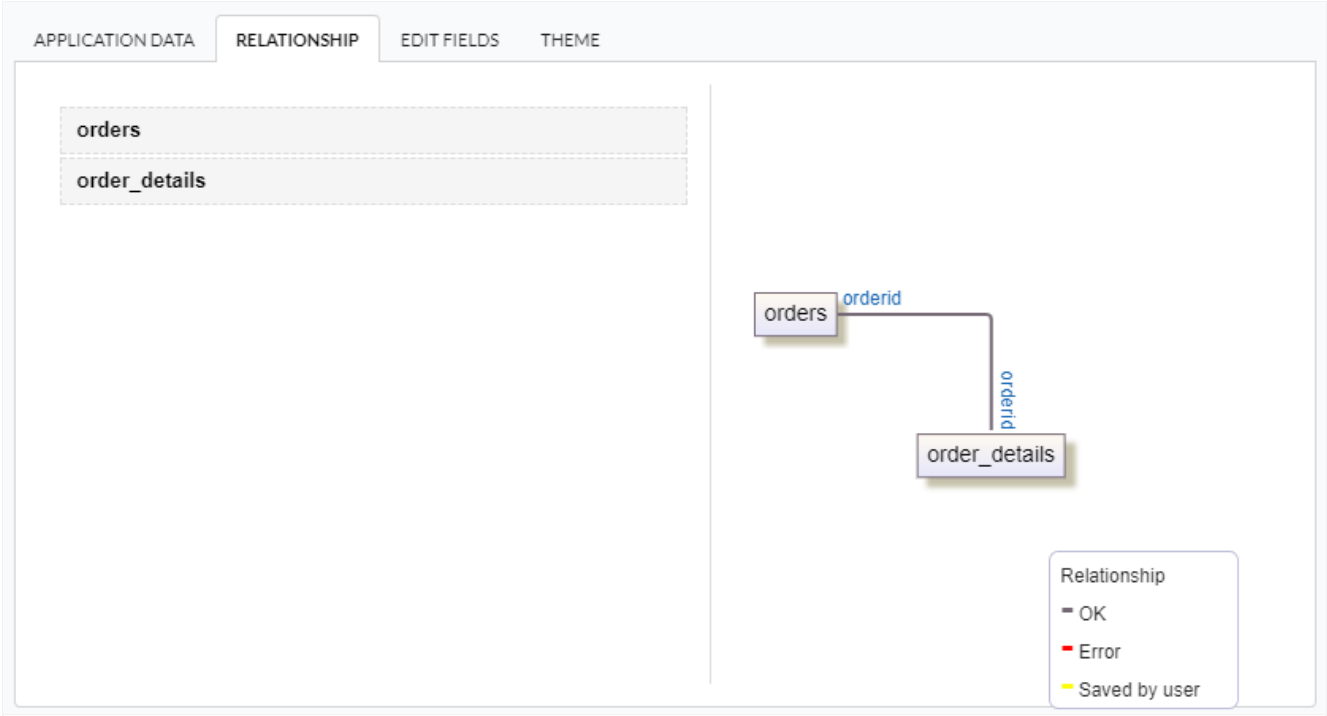
You can configure the grid to run by default by selecting the option **Run the application after creation** and clicking save.

If you deselect the option **Always show this message**, this dialog box will no longer appear.

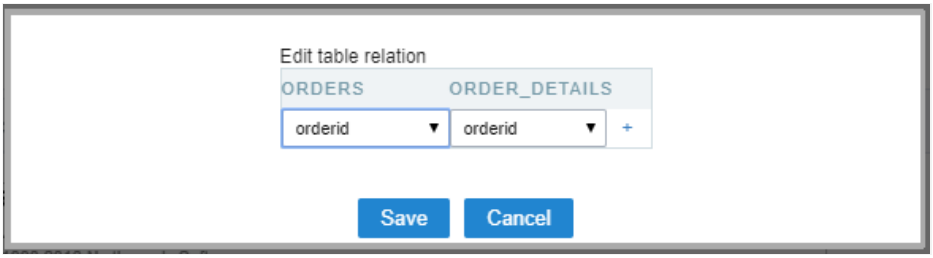
This configuration can be changed later by accessing the menu **Options > My Scriptcase**.

Relationship

When select two or more tables, the tab “Relationship” will be displayed. In this tab we can see the relationship created between the tables, where we can edit the related fields.

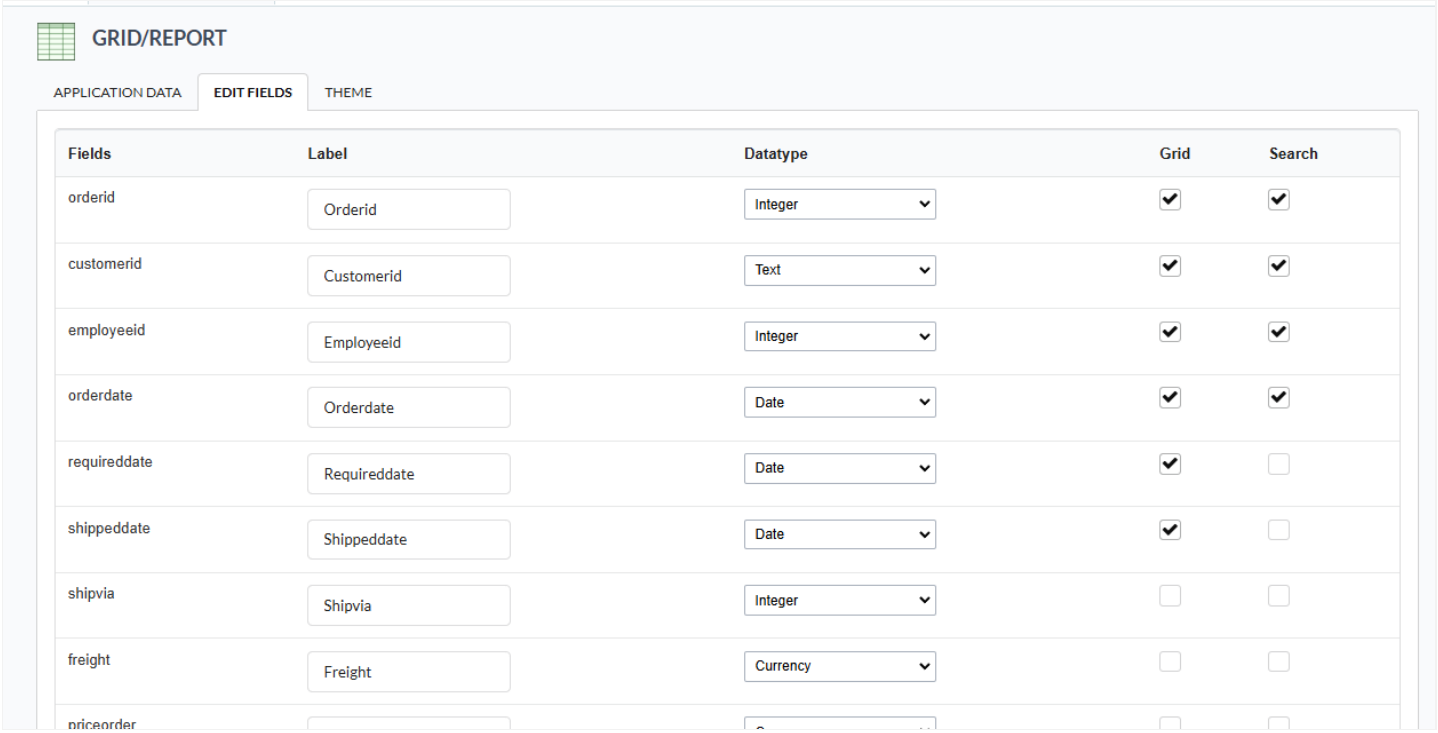


When we click in a link, in the screen above, it will displayed the related field’s edition form, as you can see in the image below.



Editing Fields

This screen displays the fields of the selected tables and allows adjustments to be made before creating the application, such as changing the data type, display name, and other configurations.



Fields

Names of the database fields.

Label

Names of the fields in the generated application's interface.

Datatype

Specifies the field's data type.

Grid

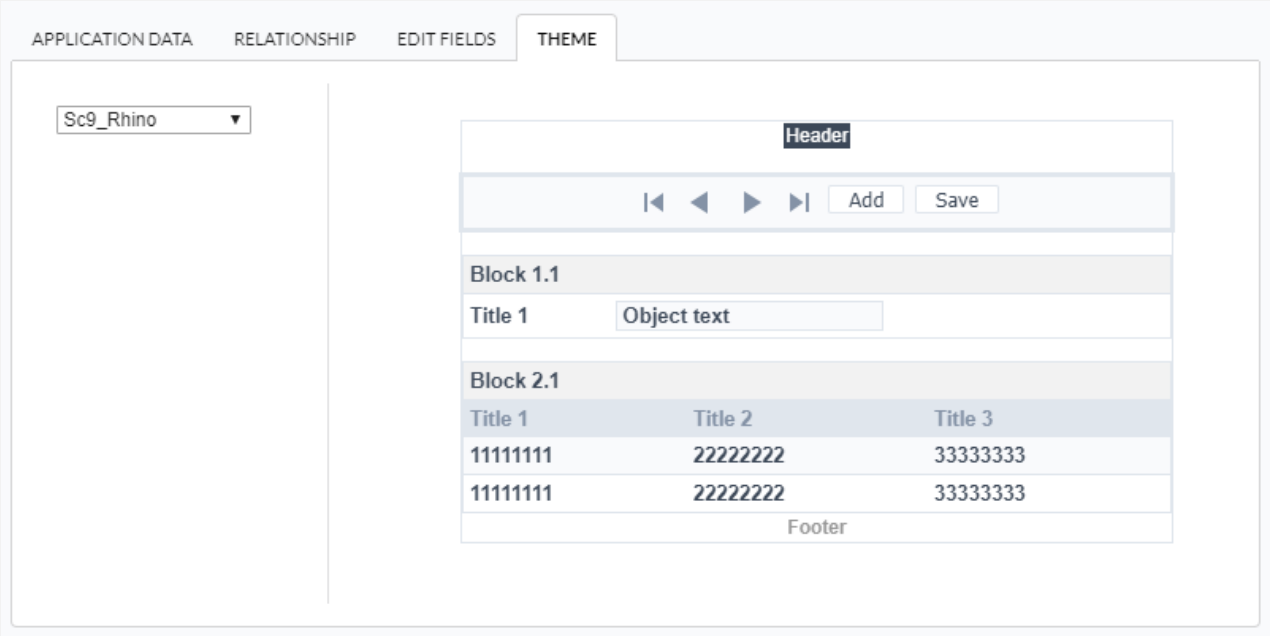
Defines the fields available in the grid.

Search

Defines the fields available in the filter.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.



Related Videos ▷

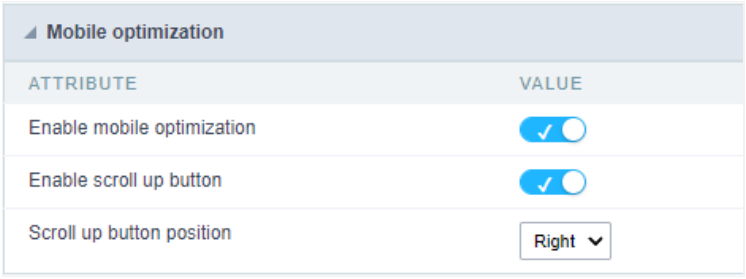
Grid Mobile Optimization

The Mobile settings allows to define the automatic optimization of generated applications to run on mobile devices.

See below the available settings.

Mobile optimization

Enable mobile optimization

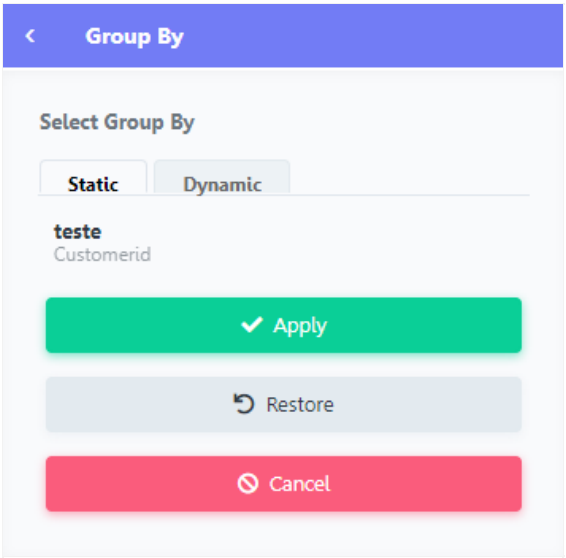


This option changes how the application HTML elements works, adapting them automatically to run on mobile devices.

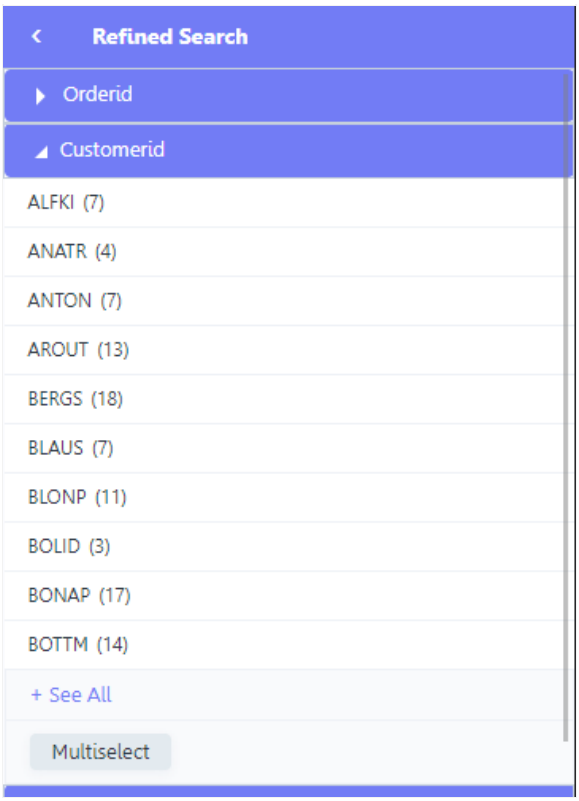
It enables this option by default when creating a new application.

See some examples of adapted screen:

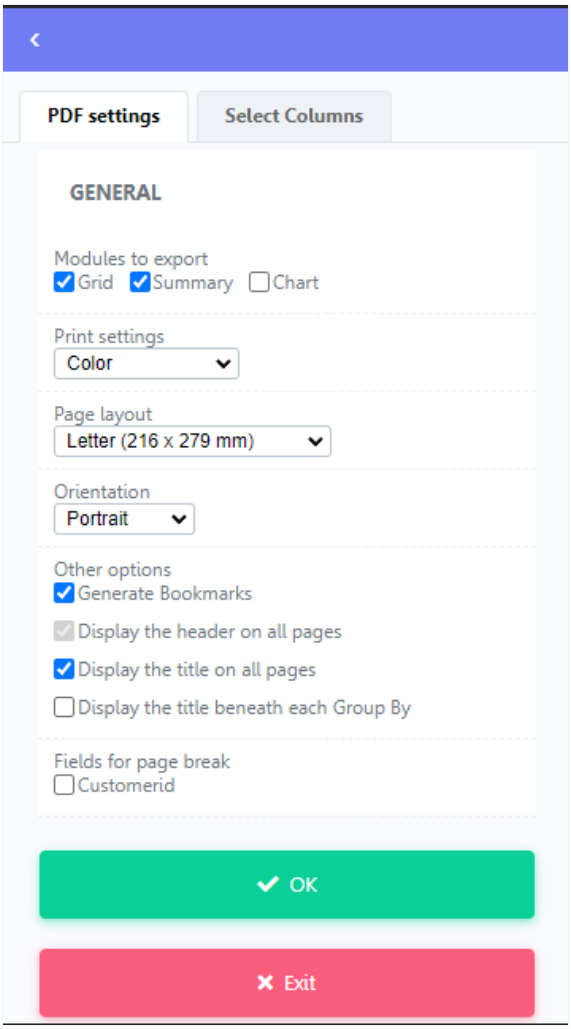
Group by Selection Screen



Refined Search Screen



PDF Export Setup Screen



Enable back to top button

It enables the displaying of the button **back to top button** in the Grid, Filter, Detail and Summary modules.

It enables this option by default when creating a new application.

Back to top button position

Defines the position of the **back to the top** button:

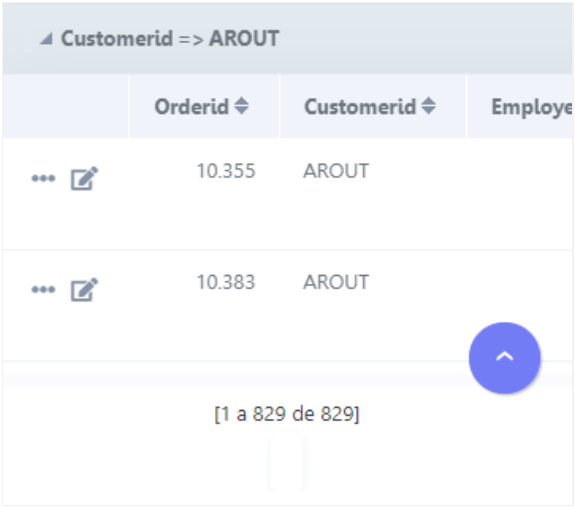
The options are:

- **Right** - Position the button in the bottom right corner;
- **Left** - Position the button in the bottom left corner;

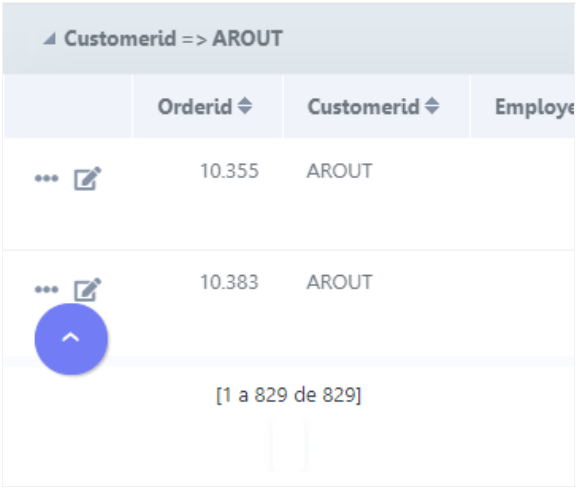
When creating a new application, the button is configured on the right by default.

See below for placement examples.

Button in the bottom right corner



Button in the bottom left corner



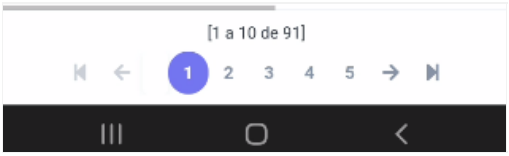
Simplified Toolbar

Defines the display format of the application's bottom toolbar, alternating between **simplified** when the option is enabled and **traditional** when these options are disabled.

To define the buttons that should compose your application's toolbar on mobile, access the **Toolbar** menu, then the **Mobile** tab.

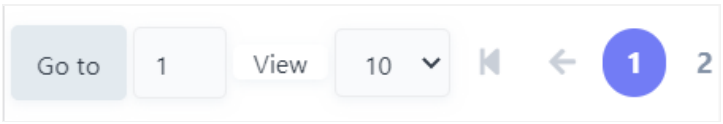
Activating it, the simplified toolbar is applied to the app by displaying only the navigation buttons and record counter as the image below.

Simple Toolbar Example



When disabling this option, the application will run with the traditional toolbar, allowing the use of the same button settings to run on the desktop or mobile devices.

Traditional Toolbar Example



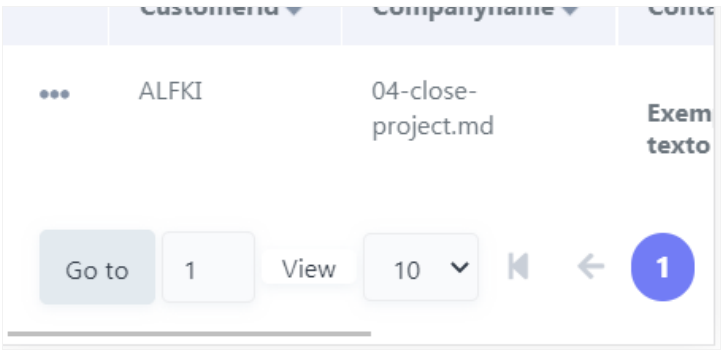
When choosing to use the traditional toolbar, the **Pin bottom toolbar** option will be enabled.

Pin the bottom toolbar

Defines the behavior of the lower toolbar when running the application on mobile devices with the **Simplified toolbar** option disabled.

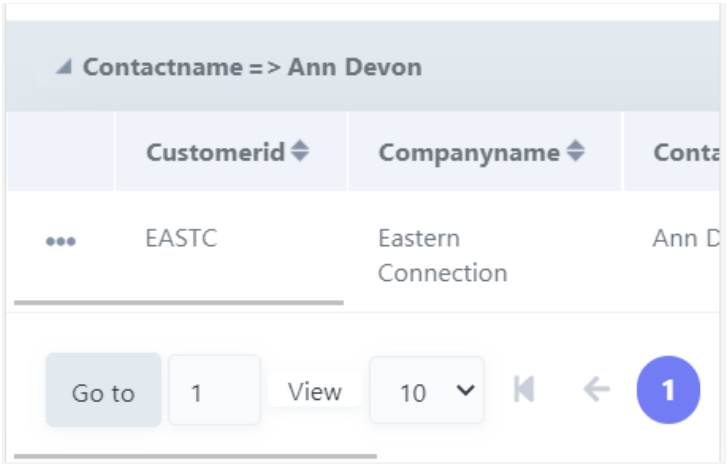
When enabling these options, the lower toolbar will behave similarly to the upper toolbar, remaining fixed on the screen and thus facilitating access to the configured resources.

Example of fixed bottom toolbar



If you choose to uncheck this feature, the lower toolbar will be fixed at the end of the application, so access to the configured buttons will only be available when scrolling to the end of the page.

Example of Traditional Toolbar



This setting is available when the **Simplified Toolbar** options is turned off.

Overview

Action Bar

In the Action Bar menu, we can create new buttons (Ajax or Connection) in our grid, these will be positioned next to the records, as already occurs with the Detail buttons query detail button{:.icon} and Editing application button edit link in query{:.icon}.

This feature also allows you to change the ordering and positioning of all buttons on the action bar, including those automatically created by the tool.

> The buttons created in the action bar are only available in the query module with **Horizontal** orientation.

Visual and Button settings

See in detail how to use the action bar.

- Management and ordering - Details of the initial screen of the **action bar** menu.
- Link Button - Details of creating and using the button.
- Ajax Button - See how to create and use the button.
- Edit bar Visual - General editing of action bar buttons display.

Action Bar Macros

Macro	Ajax Button	Link Button	Description
sc_actionbar_clicked_state			Returns the current state of the ajax button created in the action bar.
sc_actionbar_state			Changes the current state of the button to a new state.
sc_actionbar_disable			Disables action bar buttons.
sc_actionbar_enable			Enables action bar buttons, disabled by sc_actionbar_disable macro.
sc_actionbar_hide			Hides an action bar button.
sc_actionbar_show			Show action bar buttons, hidden by sc_actionbar_disable macro.

Links Relacionados

- sc_actionbar_state
- sc_actionbar_clicked_state
- sc_actionbar_disable
- sc_actionbar_enable
- sc_actionbar_hide
- sc_actionbar_show

Button List

On this screen, all the action buttons of the application will be listed.

Button ordering and placement

Both the ordering and the positioning of the action bar buttons will be carried out on this screen using dropdown. Just keep the left mouse button pressed over the desired button and position it in the desired location.

Buttons can only be positioned at the ends of the grid, either on the left side or on the right side.

You can also sort the **detail button** and the **application edit button** .

Column of Actions

 In this column we have some access links for the individual configuration of the buttons.

Configuration options by button type

Configuration link	Link button	Ajax button	Description
properties			Access the screen link properties from the link button.
Link			Screen of link configuration, allowing to redo the connection or to add some new parameter in the existing connection.
Ajax			Access the button ajax event created.
Visual			Access the screen **button visual**. In this screen we will have the individual configurations of the appearance of the buttons.
Delete			Deletes the button and all its settings. When deleting an Ajax button the event will also be deleted

Create New Button

Accesses the button creation screen, which can be of all types: **Link** and **Ajax**

Edit Bar Visual

This button opens the general editing screen for the visual of the action buttons. Click here for more details on editing the visual

Creating Link button

The **action button** of the *link type* allows you to create a link between the source application (grid) and any application in the project (destination application). This link will be activated in the onclick event of the button.

> **Source Application** is used to identify the application where the link will be created.The Source application will have the call to the **Destination Application**, which refers to the application that will be called after activating the connection.

Example of the action button in the application

See below how to create and configure this type of button.

Creating a link button

on the screen of list of buttons click **Create New Button**.

Button data

Then, select the **Type** and inform the **name** of the button.

> The **name** field only allows numbers, unaccented letters and underscores.

Button visual settings

On this screen you must define the appearance of the button for displaying in your application.

Check out the detailed view configurationoptions below.

Display type

This attribute defines the display format which can be: **FontAewsome**, **Button**, **Image** or **text**.

> Some display options are displayed according to the **type of display** used.

The options are:

FontAwesome

This is the default value of the attribute. In this setting, the button will be shown only how a font awesome icon.

See a font awesome example

Button

In this configuration, the button will be shown as a standard Scriptcase button, inheriting the button theme used in your application.

After selecting this option, another option (**Button format**) will be displayed and you will have to define how the button label will be displayed. In this attribute you can choose between:

- **FontAwesome** - The button will only display with FontAwesome.
- **Text** - The button will only display with text.
- **Font Awesome and text** - The button will appear with FontAwesome and text, with the text positioned to the right.
- **Text and Font Awesome** - The button will appear with FontAwesome and text, with the text positioned to the left.

See a button example

Image

Using this option, you must upload or select an image in the Scriptcase image manager.

Example of action button as image

Text

Using this option, the action bar button will be display how a link.

Example of action button as Text

ID

This attribute defines the name to a button state.

> **No allowed** the use langs or anytype variables in this attribute.

Font Awesome

Defines the font awesome that will be displayed in the button.

> This option is displayed when selecting in the **Type of Display** attribute the valures: *FontAwesome* or *Button*

Action button example as Font Awesome image

Image

Defines the image that will be displayed in the application when selecting **Type of display** as *image*.

Click on the icon to open the [image manager][link_gerenciador_img] {:target="_blank"} and select the desired image.

Text

The **text** attribute define text that will be display how a link in the action bar.

> This option is displayed when selecting in the **Type of Display** attribute the valures: *Button* or *Text*

The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.

> See more details about langs and about our Data dictionary.

Color

Sets the color of the Font Awesome icon. Cif no value is informed, the icon assumes the color defined in the general configuration of the action bar visual.

> Displayed only when using **Type of display** as *FontAwesome*

Color (Hover)

Sets the color of the Font Awesome icon in hover state, when hovering the mouse over the icon.

Available when selecting **Display Type** as *FontAwesome*.

Color (Active)

Sets the color of the Font Awesome icon in the active state.

Available when selecting **Display Type** as *FontAwesome*.

Button Visual

This attribute defines the color of the button. You can select a color from the list or use the *Theme* option to inherit the button layout used in your application.

Only available when selecting **Display Type** as *Button*.

Button Visual (Hover)

This attribute sets the color of the button on button hover. You can select a color from the list or use the *Theme* option to inherit the button layout used in your application.

Only available when selecting **Display Type** as *Button*.

Hint

Defines the help text for the button.

The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.

See more details about langs and about our Data dictionary.

Confirmation Message

The confirmation message is displayed after clicking the button and before that action is performed.

Confirmation Message example

The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.

See more details about langs and about our Data dictionary.

After making the visual settings, click on **Save button and configure link**. The button settings will be saved and you will be redirected to the link creation screen to define the application and target application behavior.

Application List

All project applications available for the link will be listed. In this step, the developer must select the destination application that will be used in the connection.

Finding Applications

To assist in locating the application, you can use the **grouping buttons** or the **Search** field.

Search field

The search is performed by application name.

Grouping Buttons

In this grouping there are three visualization options:

All

Lists all applications in the project alphabetically **a-z**.

Example of applications without clustering

By folder

Lists applications grouped by their folder, according to the organization of applications in the Explorer Project.

Example of applications grouped by Folder

- **Item 1** - List of folders created in Explorer Project. Select the folder and all applications in the folder will be displayed on the **item 2**.
- **Item 2** - List of applications in the folder selected in the__Item 1__.

By type

Lists applications grouped by their respective type.

Example of applications grouped by type

Item 1 - Types of existing applications in Scriptcase. Select the type of application you want to list, the applications corresponding to the selected type will be listed in **item 2**. **Item 2** - List of applications referring to the type selected in **item 1**.

When selecting the destination application, click on **Next »**.

Parameters Definition

In this step, the developer must inform the values that will be passed to the parameters of the destination application.

These parameters can be:

- Global Variables - Which can be defined in the events or in the target application's SQL.
- Primary Key - For connections created with applications of the types: Form and Calendar.

Parameters

Lists all parameters defined in the target application.

Type

Defines the source type of the value that will be sent to the parameter defined in the destination application.

The options are:

- **Fields** - It uses the value of a field from the source application as a parameter.
- **Variable** - It uses the value of a global variable, defined in the source application, as a parameter. **This option will only be listed if a global variable is defined in an event in the source application**.
- **Fixed value** - It uses a fixed value, defined in the value column, as a parameter. **In this option only alphanumeric values are allowed**.
- **No value** - Using this option no value is passed. When using this option as a parameter for a primary key, the target application will be displayed in include mode.

Value

Defines the value that will be sent as a parameter. The options in this column change according to the selected **Type**.

- When selecting **Campo** - The value column will list all the fields of the source application, which will send the value.
- When selecting **Variable** - All global variables defined in the source application will be listed.
- When selecting **Fixed value** - A field will be displayed for the value to be informed. The use of variables is not allowed in this option and we must use only alphanumeric values.
- When selecting **No Value** - In this case, no value will be sent to the parameter.

Refresh Button

This button enables ajax reloading of the list of parameters, in this way it is possible to change or add a parameter in the destination application without the need to restart the creation of the connection in progress.

Link Properties

These properties define the behavior of links, such as opening location and return URL.

Link operation mode

Defines how the target application will be displayed on the link. This attribute is also responsible for defining the available configurations according to the selected link operation mode. Check the Table below.

Open in the same window

The destination application will be displayed on the same screen as the source application. With this display option, the exit button of the destination application must have the behavior of returning to the previous application.

Open in another Window

The target application will open in a new browser window. With this display option, the target application’s exit button should close the open window.

Open in another tab

The target application will open in a browser tab. With this display option, the exit button on the target application should close the open tab.

Open in Iframe

The destination application will be displayed in an iframe next to the source application.

open in parent

The target application will be displayed in the parent of the source application.

> If used in a security module, this option displays the destination application on the same menu screen, overlapping the application and keeping the session logged in at login.

Modal

The target application will be displayed in a modal. With this display option, the exit button on the target application should close the open tab.

Link Properties attributes available according to selected Link Operation Mode

Link Attributes and Modes of Operation	Open in the same window	Open in another Window	Open in another tab	Open in Iframe	open in parent	Modal
Called application exit URL						
Closes the form after an update						
Close form after insertion						
Open the application in a tab within the menu						
Title for Tab						
Tab hint						
Active Tab Icon						
Inactive Tab Icon						
Width						
Height						

Exit URL for the target application

This attribute defines to which page or other application of the project the target application of the link will return. This return can be triggered through the back button or through some behavior configuration performed in the link, for example, the **Closes the form after an update** option (This attribute is exclusive for links with forms).

It is possible to define an external URL, for example https://scriptcase.net, or a project application where it will be necessary to inform only the application name.

> If no value is informed, the return will be to the source application itself. This option is only available when the Link Operation Mode is set to: Open in same Window or Open in parent.

Close form after update

This attribute defines the behavior of the form when performing a record update from a link.

When you check yes, the form opened in the connection will be closed automatically when updating the record. If the attribute is marked as **não**, the destination application will remain open after the update, allowing new registry changes to be made.

> This attribute is available only when the target application is a **form**.

Close form after insert

![Define se o formulário chamado será fechado após realizar inserção de algum registro][img_att_fechar_no_insert]

This attribute defines the behavior of the form when inserting a record from a connection.

When you check yes, the form opened in the connection will be closed automatically when inserting a record. If the attribute is marked as **not**, the target application will remain open after inserting records, thus allowing new records to be inserted. [img_att_fechar_no_insert]: /assets/images/docs/app/comum/ligacao/geral/atributos_propriedade/att_fechar_no_insert.png

> This attribute is available only when the target application is a **form**.

Open the application in tab within the menu

This attribute defines the connection behavior when the origin application is accessed from a menu application, which in turn must be configured with the opening of items in tabs. See how to enable this feature in the applications of menu and tree menu.

> The attribute is available when using the *Link operation mode* defined as: **Open in another window** or **Open in another tab**.

When activated the destination application will open in a tab within the menu application. **When checking no** the target application will be opened in a **new window** or a **new browser tab**, according to the selected *Link operation*

mode.

Tab Title

Allows the developer to define a menu tab title where the target application will be displayed. If no value is informed, the tab will inherit the name of the called application.

> This setting is available if the attribute **Open the application in a tab within the menu** is set to *Yes*
> The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.
>
> See more details about langs and about our Data dictionary.

Tab Hint

Allows the developer to define a text to guide the end user in using the system. Help text will be displayed when hovering the mouse over the menu tab.

> This setting is available if the attribute **Open the application in a tab within the menu** is set to *Yes*
> The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.
>
> See more details about langs and about our Data dictionary.

Example of Hint in menu tab with destination application

Icon for the Tab when active

Defines the image that will be displayed on the menu tab when activated. If no image is informed, the tab will inherit the default image, defined in the menu icons.

> This setting is available if the attribute **Open the application in a tab within the menu** is set to *Yes*

Example of active tab icon

Icon for the Tab when Inactive

Defines the image that will be displayed on the menu tab when activated. If no image is informed, the tab will inherit the default image, defined in the menu icons.

> This setting is available if the attribute **Open the application in a tab within the menu** is set to *Yes*

Example of the inactive tab icon

Width

> This attribute is available for configuration only when using the **Link operation mode** as *Modal*.

Defines the width of the modal where the form will open. This width refers to the horizontal space available for displaying the application.

This attribute obligatorily uses **pixels** as the measurement unit, therefore, only numbers must be informed to define the width. The width of the application displayed in the modal respects the definition of the **Table Width** attribute.

Example of the application opened in the modal

Height

This attribute is available for configuration only when using the **Link operation mode** as *Modal*.

Defines the height of the modal where the form will be displayed. This height refers to the vertical space available for displaying the application.

This attribute obligatorily uses **pixels** as the measurement unit, therefore, only numbers must be informed to define the width. The height of the application is calculated automatically and respects the number of fields selected for the application.

Example of the application opened in the modal

Form Properties

These properties are only available in links where the destination application is a form, this configuration block defines whether or not to display the **Update** (*Insert*, *Update* and *Delete*) and **Navigation** (*first*, * previous*, *next* and *last*) of the target form, according to the toolbar settings of the called application.

It is important to point out that the attributes in the Form Properties block do not override the toolbar settings of the called form, they only determine whether the buttons configured there will be displayed or not in the link.

> This attribute is not available if the **Link Operation Mode** attribute is set to *Open in an iframe*. In this configuration, the form's buttons will be displayed on the toolbar of the link's source application. Click here for more details on the query toolbar.

Enable Insert Button

Sets whether or not to display the **Insert** button on the target form. For this, it is necessary that the button *include* is selected to be displayed in the settings of the toolbar of the appplication.

Enable Update Button

Sets whether or not to display the **Refresh** button on the target form. For this, it is necessary that the button *change* is selected to be displayed in the settings of the toolbar of the appplication.

Enable Delete Button

Sets whether or not to display the **Delete** button on the target form. For this, it is necessary that the button *Delete* is selected to be displayed in the settings of the toolbar of the appplication.

Enable Navigation Buttons

Determines whether or not to display the **Navigation Buttons** (*First*, *Back*, *Page Navigation*, *Next*, *Last*) configured in toolbar of the target form.

Keep the WHERE clause in the target application

> This attribute is only available when enabling navigation buttons.

This attribute makes it possible for the WHERE clause of the source application (Query) to be maintained in the destination application. This means that the filters performed in the query will be inherited by the target form, enabling better quality in the records displayed for editing.

Grid Properties

Grid properties allow changing the behavior of the target application, without having to change the application.

It is important to point out that these configurations do not force the configurations in the destination application, they only change the previously made configurations. For example, if the target application has the filter module disabled, using **Initial mode** as *filter* in the connection will not have any effect.

> These properties are only available on connections where the target application is a grid.

See below for more details on the attributes.

Initial Mode

Defines the opening module of the target application, which can be **Grid** or **search**. For this feature to work, the target application must have the modules enabled.

Number of Lines:

Defines the number of lines that will be displayed in the target application. If no value is informed, the configuration of the called application will be respected.

Number of Columns

Limits the number of columns displayed in the target application.

> The entered value must be less than the number of columns displayed in the target application.

Paging

This attribute defines the type of pagination used in the application, which can be **Partial** where the developer defines the number of lines per page or **total** which displays all records on the same page.

Partial

Displays the target application with a limited number of records per page. The number of records per page can be defined in the **Number of Lines** attribute, if no value is informed, it will inherit the target application's configuration.

> This attribute does not force partial pagination on the target application, only the navigation buttons configured on the toolbar.

Example of toolbar with partial pagination

Total

Forces the target application to display without paging, displaying all records on a single page. In this case the navigation buttons are removed.

Example of toolbar with full pagination

Display Header

Defines whether the target query header will be displayed. For the feature to work correctly, the application and destination must have header options configured.

Active Navigation Buttons

Determines whether or not to display the **Navigation Buttons** (*First*, *Back*, *Page Navigation*, *Next*, *Last*) configured in toolbar of the target application.

Iframe properties

This configuration block is available when using the **Link Operation Mode** set to *Open in an iframe*.

Position of the iframe relative to the main application

Defines the positioning of the iframe relative to the source application.

- **Below**: The iframe will be displayed below the source application.
- **Above**: The iframe will be displayed above the source application.
- **Right**: The iframe will be displayed on the right side of the source application.
- **Left**: The iframe will be displayed on the left side of the source application.

Iframe height

Defines the height of the iframe where the form will open. The attribute obligatorily uses pixels as the unit of measurement, therefore, we must inform only numbers in its definition. If no value is informed, the height will be based on the connection origin application.

Iframe width

Allows the width of the iframe to be set.

Defines the width of the iframe where the form will open. The attribute compulsorily uses pixels as the measurement unit, thus, only numbers must be informed to define the width.

Creating Ajax button

The **action button** of the *ajax* type creates an ajax event of the onClick type, allowing the execution of routines without the need to reload the application.

Example of the action button in the application

See below how to create and configure this type of button.

Creating one button

on the screen of list of buttons click **Create New Button**.

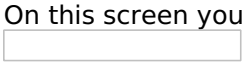
Button data

Then, select the **Type** and inform the **name** of the button.

> The **name** field only allows numbers, unaccented letters and underscores.

Button visual settings

On this screen you must define the appearance of the button for displaying in your application.



Check out the detailed view configurationoptions below.

Display type

This attribute defines the display format which can be: **FontAewsome**, **Button**, **Image** or **text**.

> Some display options are displayed according to the **type of display** used.

The options are:

FontAwesome

This is the default value of the attribute. In this setting, the button will be shown only how a font awesome icon.

See a font awesome example

Button

In this configuration, the button will be shown as a standard Scriptcase button, inheriting the button theme used in your application.

After selecting this option, another option (**Button format**) will be displayed and you will have to define how the button label will be displayed. In this attribute you can choose between:

- **FontAwesome** - The button will only display with FontAwesome.
- **Text** - The button will only display with text.
- **Font Awesome and text** - The button will appear with FontAwesome and text, with the text positioned to the right.
- **Text and Font Awesome** - The button will appear with FontAwesome and text, with the text positioned to the left.

See a button example

Image

Using this option, you must upload or select an image in the Scriptcase image manager.

Example of action button as image

Text

Using this option, the action bar button will be display how a link.

Example of action button as Text

ID

This attribute defines the name to a button state.

> **No allowed** the use langs or anytype variables in this attribute.

Font Awesome

Defines the font awesome that will be displayed in the button.

> This option is displayed when selecting in the **Type of Display** attribute the valures: *FontAwesome* or *Button*

Action button example as Font Awesome image

Image

Defines the image that will be displayed in the application when selecting **Type of display** as *image*.

Click on the icon to open the [image manager][link_gerenciador_img]{:target="_blank"} and select the desired image.

Text

The **text** attribute define text that will be display how a link in the action bar.

> This option is displayed when selecting in the **Type of Display** attribute the valures: *Button* or *Text*

The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.

> See more details about langs and about our Data dictionary.

Color

Sets the color of the Font Awesome icon. Cif no value is informed, the icon assumes the color defined in the general configuration of the [action bar visual][link_editar_visual]{:target="_blank"}.

> Displayed only when using **Type of display** as *FontAwesome*

Color (Hover)

Sets the color of the Font Awesome icon in hover state, when hovering the mouse over the icon.

Available when selecting **Display Type** as *FontAwesome*.

Color (Active)

Sets the color of the Font Awesome icon in the active state.

Available when selecting **Display Type** as *FontAwesome*.

Button Visual

This attribute defines the color of the button. You can select a color from the list or use the *Theme* option to inherit the button layout used in your application.

Only available when selecting **Display Type** as *Button*.

Button Visual (Hover)

This attribute sets the color of the button on button hover. You can select a color from the list or use the *Theme* option to inherit the button layout used in your application.

Only available when selecting **Display Type** as *Button*.

Hint

Defines the help text for the button.

The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.

See more details about langs and about our Data dictionary.

Confirmation Message

The confirmation message is displayed after clicking the button and before that action is performed.

Confirmation Message example

The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.

See more details about langs and about our Data dictionary.

Delete

Deletes a button state.

Botão Adicionar Estado

Adds a line for configuring a new state for ajax button.

These states can be manipulated using the macrosc_actionbar_state at ajax event.

After adding the desired states and configuring the appearance of the buttons, click on **Save button and configure ajax** to create the button and event.

Ajax Event

The ajax event will be created, which will be executed on the button’s onClick.

In this event we must code the procedures that we want to carry out. Added some macros for handling the action buttons:

- sc_actionbar_clicked_state - Returns the current state of the ajax button created in the action bar.
- sc_actionbar_state - Changes the current state of the button to a new state.
- sc_actionbar_disable - Disables action bar buttons.
- sc_actionbar_enable - Enables action bar buttons, disabled by sc_actionbar_disable macro.
- sc_actionbar_hide - Hides an action bar button.
- sc_actionbar_show - Show action bar buttons, hidden by sc_actionbar_disable macro.

Edit Bar Visual

On this screen you can customize the defined action bar buttons padding, size of Font Awesome, color of items, links and your alignment.

Padding

Sets the spacing of the element’s content to its border. This attribute applies to action bar buttons regardless of their type (*Link* or *Ajax*) or the display(*Font Awesome*, *image* and *text*).

You can use up to four space-separated values.

- One value - Applies to all edges
- Two values - Refers to the vertical and horizontal borders, respectively. *Three values - Refers to the Top, Horizontal, and Bottom borders, respectively.
- Four values - Refers to the top, right, bottom, and left edges, respectively.

Font Awesome Size

Sets the size, in pixels, of the action bar’s Font Awesome icons.

> This attribute is only applied to buttons configured with Font Awesome.

Font Awesome Color

This attribute defines the color of the Font Awesome icons created in the action bar, if no color has been defined in the individual button configuration.

> The color set individually in the button settings overrides this setting.

Font Awesome Color (Hover)

Sets the color of the Font Awesome icon in hover state, when hovering the mouse over the link.

Font Awesome Color (Active)

Sets the font awesome icon color in active state for text buttons.

Link Color

In this attribute we define the color of the text displayed in the application when using the **type of display** as *text* in the creation of the button.

> This attribute refers to action “buttons” configured as text.

Hover link color

Defines the text color in the hover state, when positioning the mouse over the link.

> This attribute is only applied to buttons configured as *text* when creating the button.

Active link color

Sets the link color in the active state for text buttons.

This attribute refers to action “buttons” configured as text.

Vertical alignment

Sets the vertical alignment of the action bar buttons.

Overwrite buttons

This attribute defines if the visual definitions, configured for Font Awesome (Padding, Vertical alignment, icon size and Color), should override the **Detail** and **Application Edit** .

Group buttons

This feature allows you to group action bar buttons side by side.

This setting is only applied to icons where the attribute **Type of display** is defined as *Button*

Example of grouped buttons

Example of ungrouped buttons

Edit Column Label

This option allows to configure a label for the application’s action bar column.

Example of a grid with the label action bar

Edit column labels

This feature allows you to set a label for the action bar column. This definition must be done differently for the left and right bar.

Label

This attribute defines the label displayed in the action bar column.

The text definition has no character restriction and can be done with a **fixed text** or using a **lang**, to create systems with multiple languages.

> See more details about langs and about our Data dictionary.

Font color

Sets the text color of the action bar column.

> If no color is defined, the text color will be defined by the theme.

Example text color

Background Color

Sets the background color of the action bar column.

> If no color is defined, the background color will be defined by the theme.

background color example

Horizontal Alignment

Defines the text placement of the action bar label.

- **Left** - Aligns the text to the left of the column.
- **Center** - Aligns the text to the center of the column.
- **Right** - Aligns the text to the right of the column.

> If no color is defined, the text color will be defined by the theme.

Example right alignment of text

Example of aligning text to center (default)

bold text

Defines if the label text will be displayed in bold or not.

Example of bold text

Example of non-bold text

Charts

This application is designed for creating dynamic charts based on SQL or Procedures.

Inside the Scriptcase Charts can be created also within the Grid application, however the application Chart has more settings options to the end user within the application generated. The idea is to apply the concept of Business Intelligence so the final system users can apply their own settings within the generated charts.

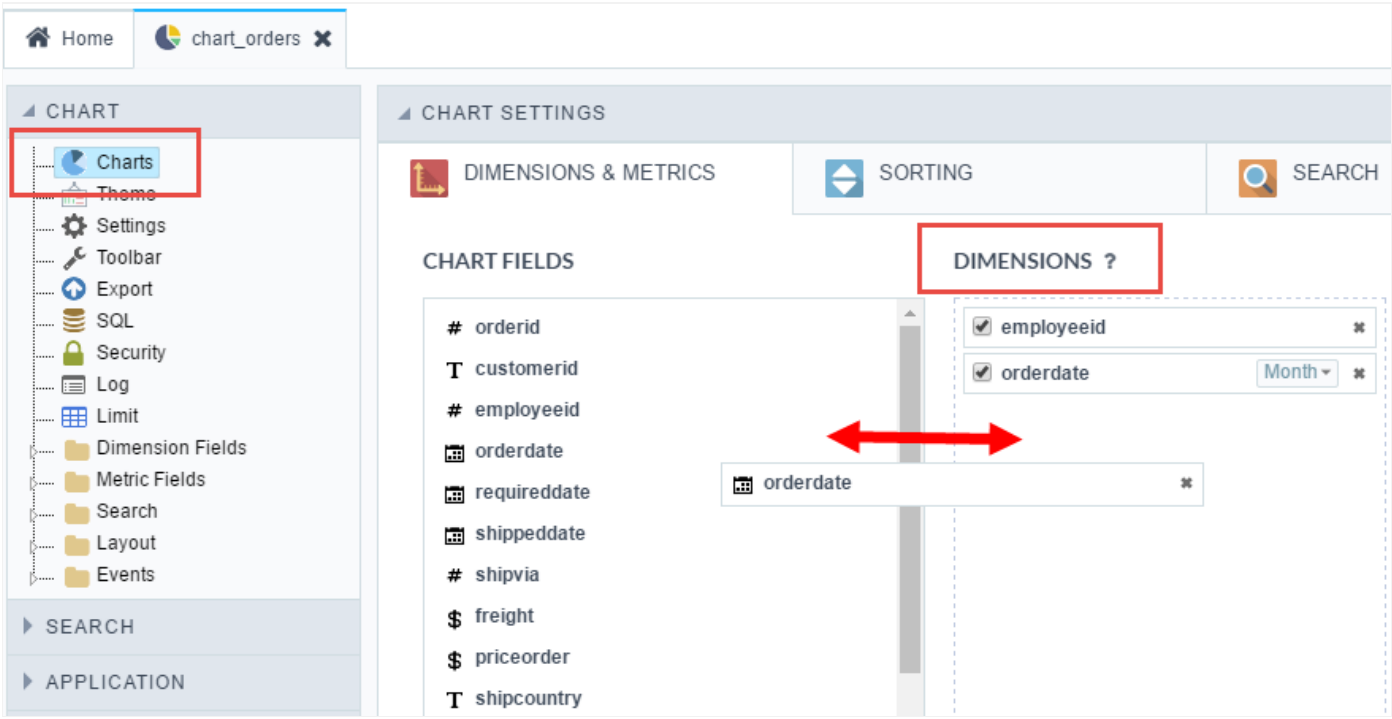
The settings within the development area uses Drag and Drop for the charts creation. You can apply dimensions and metrics dynamically using different date periods and totals.

Dimensions

Dimensions are used to group the data into categories. You can choose more than one field as a dimension, each new field as a subcategory of the previous field.

To select a field, click on its name in the “Chart Fields” area and drag it to the “Dimensions” column.

Once selected the fields, you can sort them using the drag and drop to arrange the dimensions’ final display. You can choose different periods to date fields and use the same date field more than once, with different periods. All the dimensions listed in “Dimensions” can be used by the system final user within the Chart application. You can also define whether dimensions will be displayed or not when you start the application by selecting the checkbox.



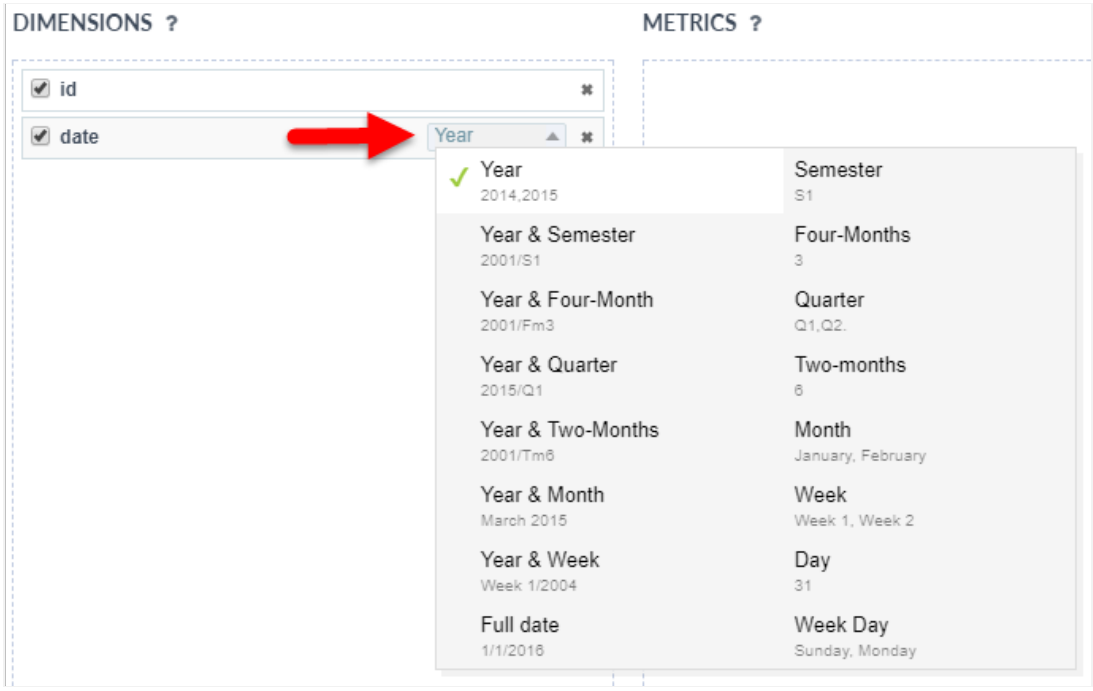


Chart dimensions. *Field settings*

Metrics

The metric is used to quantitatively view the data grouped by dimensions. Choose the fields for the chart's metrics.

To select a field, click on its name in the "Chart Fields" area and drag it to the "Metrics" column. Once selected the fields, you can sort them using the drag and drop to arrange the metrics' final display, you can also chose the total function according to the field data type.

All the dimensions listed in "Metrics" can be used by the system final user within the Chart application. You can also define whether metrics will be displayed or not when you start the application by selecting the checkbox.

The checked field value is used to set the default field just when you are not using a combination or stacked chart types, that use more than one field. If you choose more then one field Scriptcase automatically sets a value multiple chart type (Combination). You can change the initial chart inside the option "Charts' type" within the "Settings". When you check a single value (bar, line, Spline, step, Area, pie, pyramid, funnel, Radar and Gauge) you have to click and choose only one field to appear as initial value during the first generation.

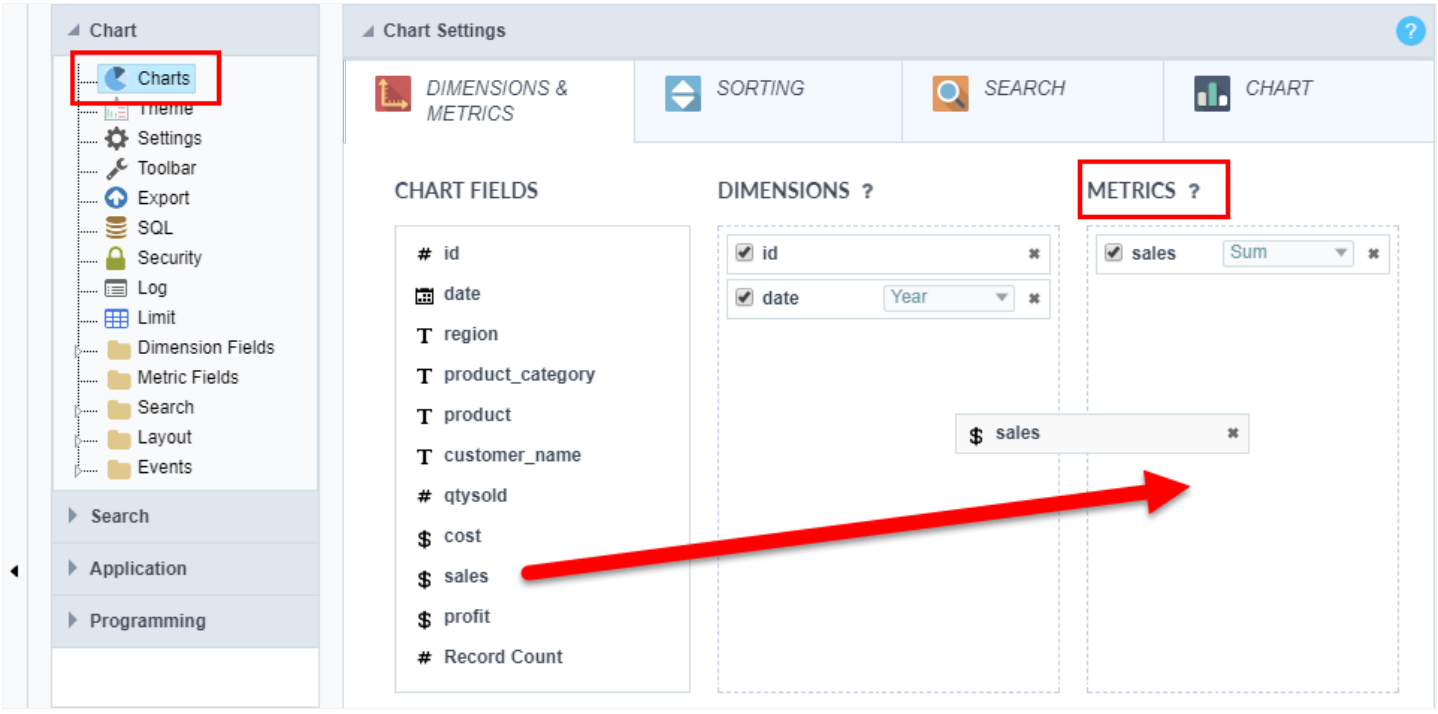


Chart Metrics.

Sorting

You can select which chart value will be ordered to start the application, and if the sort will be “ascending” or “descending”.

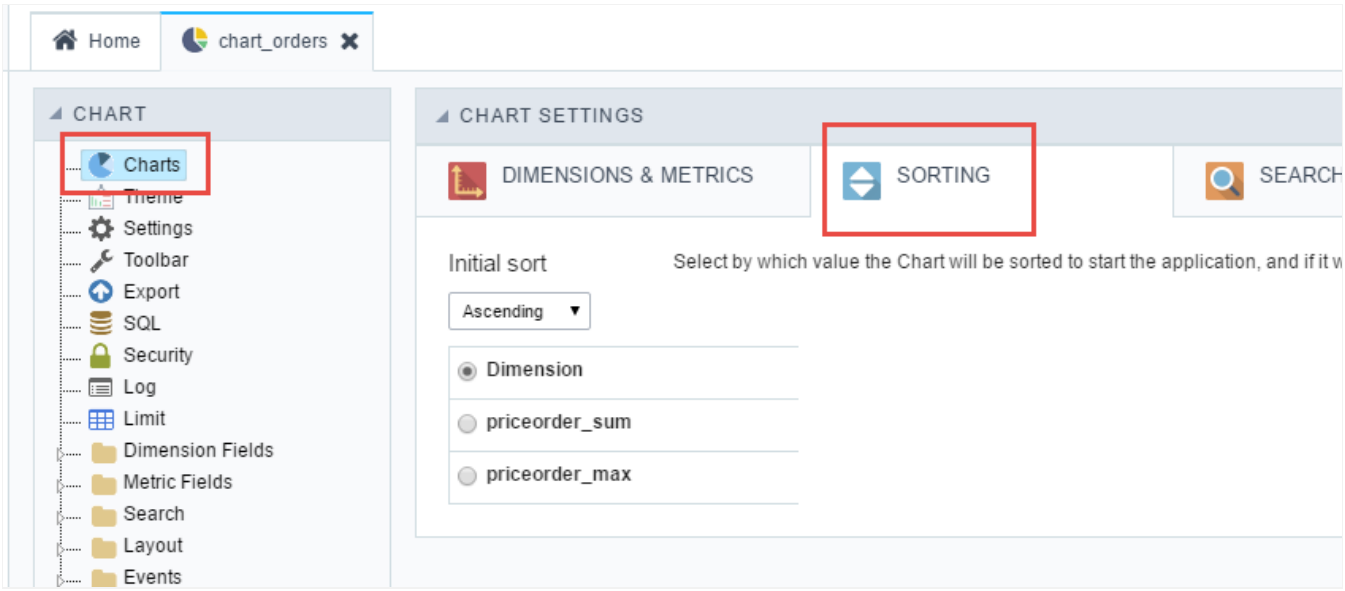


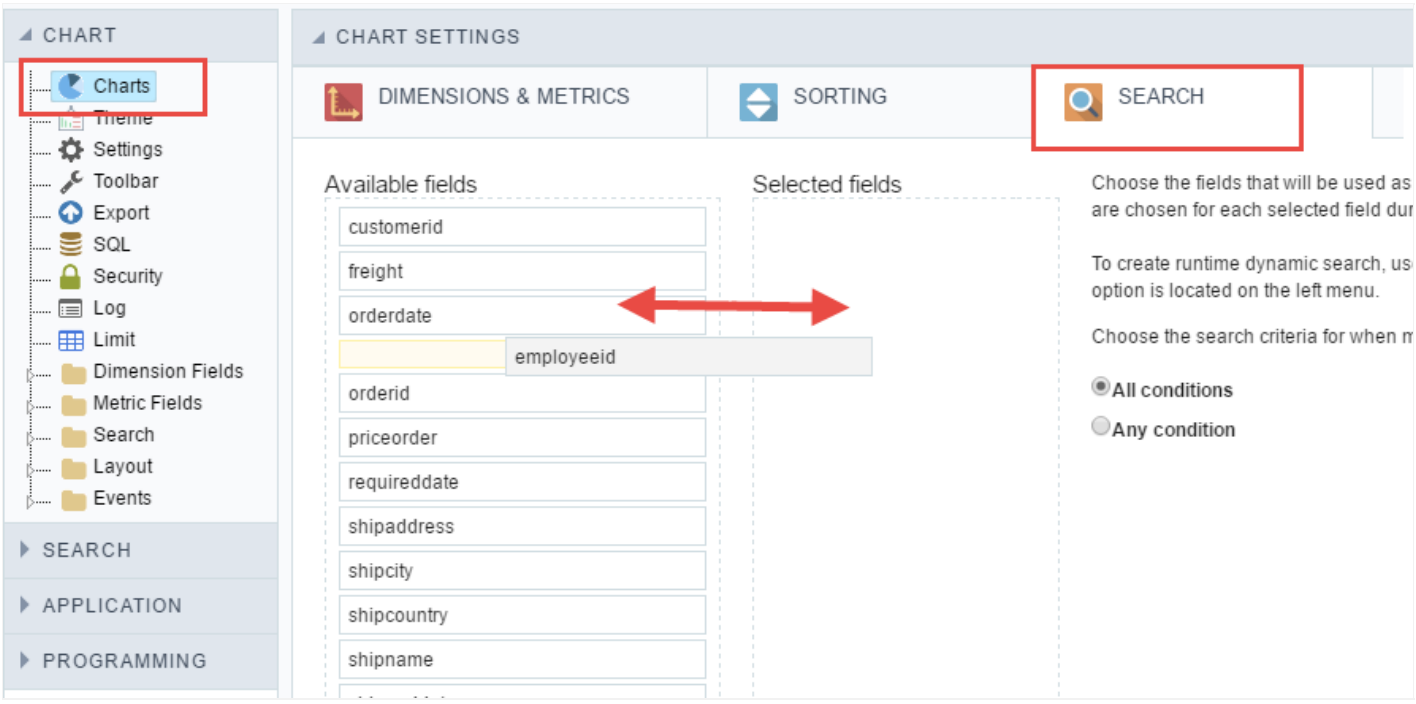
Chart sorting.

- **Initial sorting** : Using this option you can set the default sorting (ascending or descending). The system final user will also be able to customize it within the generated application.
- **Dimension** : Application starts with dimension fields sorting (ascending or descending according to the “Initial sorting”).
- **Metric** : Application starts with metric fields sorting (ascending or descending according to the “Initial sorting”).

Search

In the “Search” option you can choose the fields that will be used as static search on the chart, the application data will be initially filtered according to the settings you apply. The search values are chosen for each selected field during development, the final user will not be able to change that.

You are also able to apply other search options such as the Dynamic Search using the option “Search > Dynamic Search” (located on the left menu) and the “refined” search using the option “Chart > Search”. These two types are dynamic, it means that the end user will be able to manipulate them within the generated application.



Static search.

Chart options

In the 'Chart' tab, the developer defines the default settings for the generated chart, choosing the default chart type, color scheme, and other options. Below, each option is detailed.

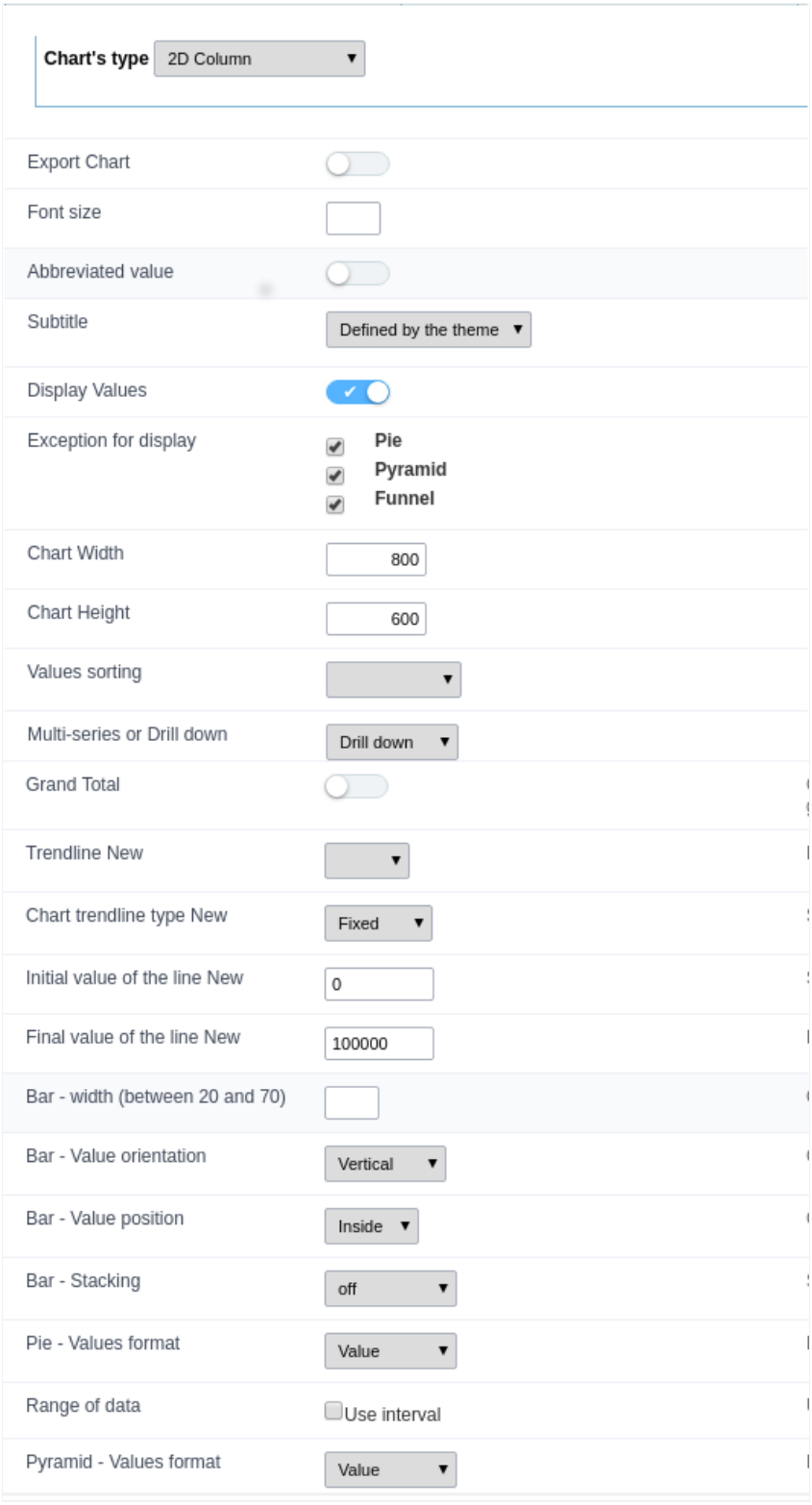


Chart Type

Defines the type of chart to be used, such as bar, pie, line, gauge, pyramid, tree map, among others.

Export Chart

Allows exporting the chart to different formats, such as **PNG, JPG, PDF, and SVG**, enabling storage, sharing, and printing of the presented data.

Font Size

Defines the font size used in the chart elements, such as titles, legends, and values.

Abbreviated Value

Determines if the values displayed in the chart should be abbreviated, such as:

- `1000` → `1K`
- `1000000` → `1M`

Legend

Shows or hides the chart legend, which identifies the data series presented.

Display Values

Enables or disables the display of values directly in the chart elements.

Exception for Display

Chart types that will not display values even when the option to show values is enabled.

Chart Width

Defines the chart width in pixels, ensuring a better visual fit.

Chart Height

Defines the chart height in pixels, allowing proper sizing.

Value Sorting

Controls the order of values displayed in the chart, which can be:

- Ascending
- Descending
- No sorting

Multi-Series or Drill Down

Allows defining whether the chart will use multiple data series or a Drill Down structure, enabling navigation between data levels.

- **Multi-Series**: Displays multiple data series in the same chart, allowing the comparison of different datasets.
- **Drill Down**: Enables data interaction, allowing the user to click on a chart element to view more detailed information at a new level.

Expanded Tooltip

Defines whether the tooltip (information displayed when hovering over elements) will expand to show more details.

Adaptive Y-Axis

Automatically adjusts the Y-axis scale according to the values displayed in the chart.

Grand Total

Adds a totalizer value to the chart, displaying the sum of the presented values.

Trend Line

Adds a trend line to the chart, helping to identify patterns and trends in the data.

Can be configured as:

- **Line**: Displays the data trend in line format.
- **Area**: Fills the area below the trend line, visually highlighting data variations.
- **No Value**: No trend line will be applied to the chart.

Trend Line Type

Defines the type of calculation applied to the trend line.

Note: This option is only displayed when the trend line is set as **Line**.

Can be configured as:

- **Fixed**: Displays a straight line at a predefined value.
- **Average**: Calculates and displays the line based on the average of the chart values.
- **Linear**: Applies a linear calculation to indicate the data trend over time.

Initial Line Value

Defines the numerical value where the trend line will be positioned when using the **Fixed** type.

This value represents a reference limit within the chart, aiding in data analysis.

Bars - Width (Between 20 and 70)

Controls the width of the bars in a bar chart, ranging from 20% to 70% of the available space.

Bars - Value Orientation

Defines whether the values in the bars will be displayed vertically or horizontally.

Bars - Value Positioning

Determines the position of values inside the bars:

- Inside
- Above

Bars - Stacking

Defines how bars are stacked in the chart.

- **Off**: Follows the default setting, displaying bars separately.
- **On**: Displays a standard stacked chart where values are summed vertically.
- **Percentage**: Generates a stacked chart where bars occupy **100%** of the available area, distributing values proportionally.

Pie - Value Format

Defines how values will be displayed in the pie chart, which can be:

- Absolute values
- Percentage (%)

Data Range

Available only for **Gauge** charts.

When enabled, a list of fields is displayed where the following must be defined:

- **Base Value**: The starting point for measurement.
- **Range**: The value range to be considered.
- **Color for each range**: Defines the color for each interval in the chart.

This configuration highlights different value zones within the Gauge, facilitating visual analysis.

Gauge - Show Scale Marks

Enables or disables the display of scale marks in the **Gauge** chart.

Pyramid - Value Format

Defines how values will be displayed in the **Pyramid** chart, as **Absolute value** or **Percentage**.

Tree Map - Title

Defines the main title of the **Tree Map** chart, improving data identification.

Related Links 🔗

Related Videos ▷

Chart Settings

Chart settings

In the chart Setup screen the user can change some application viewing options, which charts will be displayed, the labels, margins and alignments. Below detailing each of the available attributes

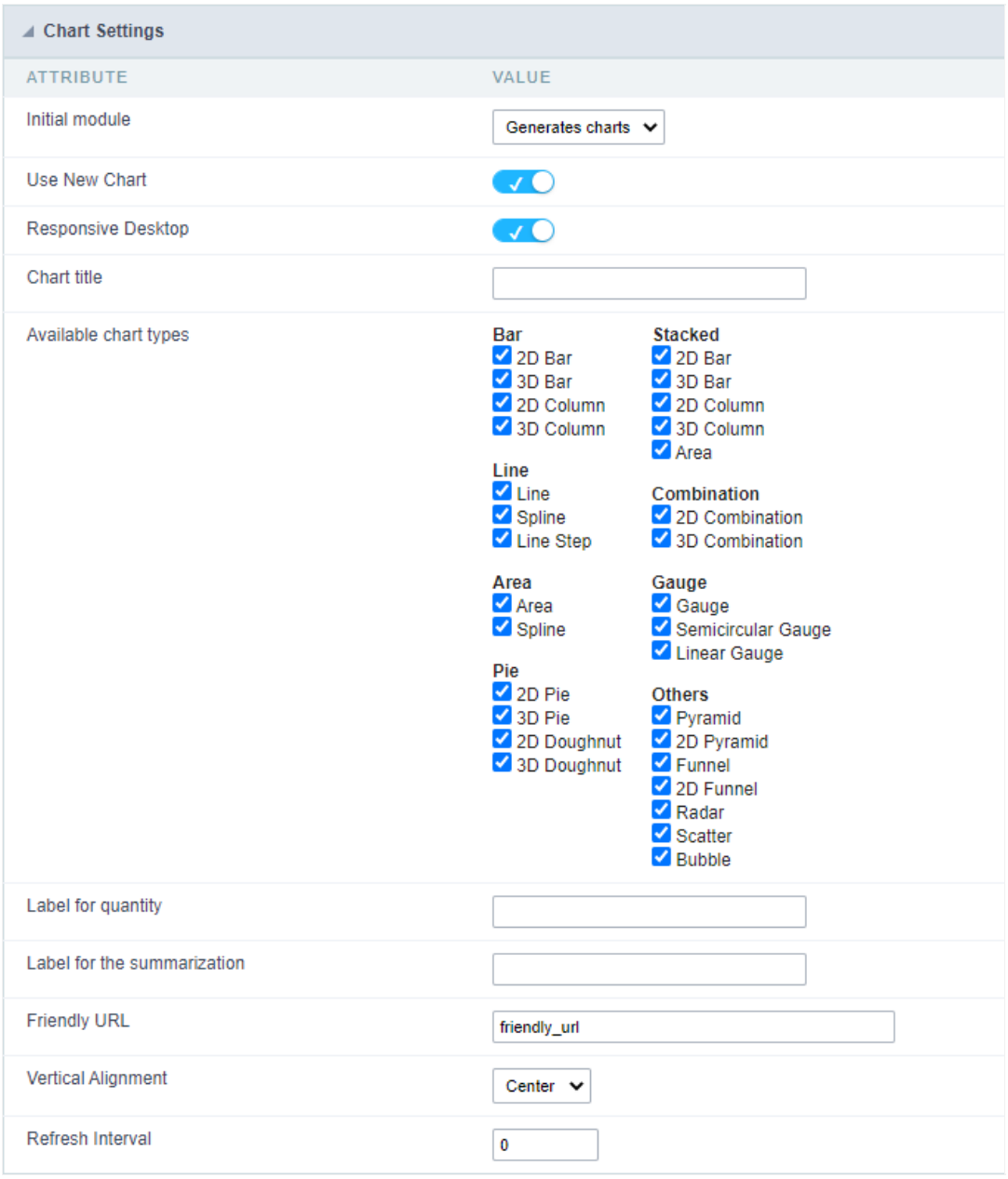


Chart settings interface

- **Responsive Desktop** : By checking this option the Chart will automatically adapt itself to the user's screen according to the size of the browser.
- **Chart title** : This option allows you to set a title for the chart.
- **Available chart types** : List of available chart types for user's choice within the final application via the button "Chart Types". The graphics checked will be available, if the developer does not want to display a chart type must do not check them in this interface.
- **Label for quantity** : Label the y-axis for generic values of the record count.
- **Label for summarization** : Label the y-axis for generic values of the summary functions - sum, average, weighted average, maximum and minimum.
- **Friendly URL** : This field allows you to change the URL that will be called by the application. Allowed characters are the same available on URLs: a-z, A-Z, 0-9, -_. This option can also be changed on the home screen, on the "Friendly URL"

column at the applications list.

- **Horizontal Alignment**: It allows to define the horizontal alignment of the application (Centered, Left, Right).
- **Vertical Alignment**: Allows you to define the initial vertical alignment of the application (Above, Centered and Below).
- **Margins** : Sets the application margins in pixels (up, down, right and left).
- **Refresh Interval** : Allows you to set a reload interval for the page, in seconds. When is set as zero, there will be no page reload.

Related Links 🔗

Related Videos ▷

Chart Toolbar

The application toolbar has two segments: Top and Bottom, in a way that is possible to define to display buttons into both areas. Those areas work independently, allowing them to display the same button, for example.

It's also possible to select the buttons and their position if the application is running on a mobile device.

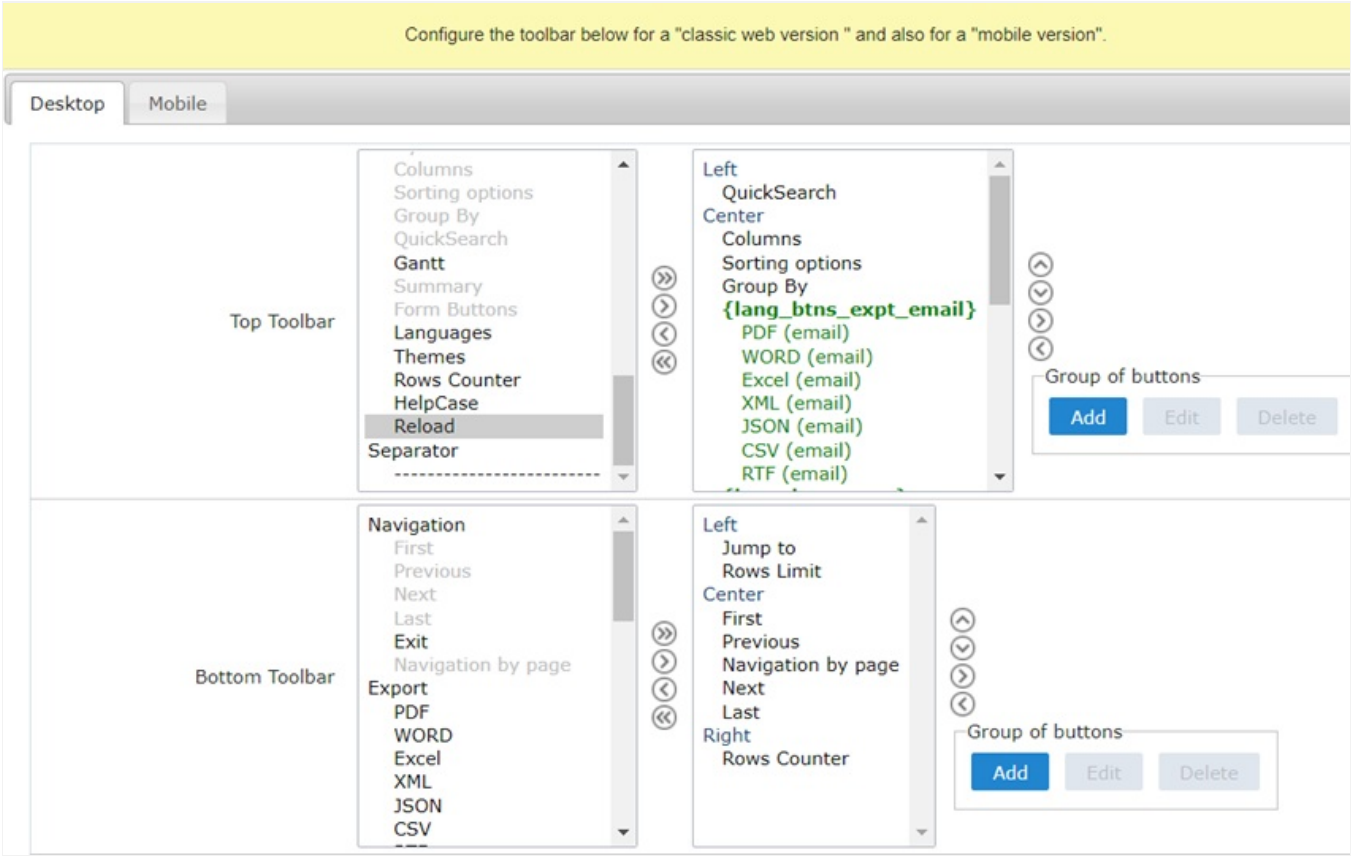
Toolbar

Desktop

Here we must inform the toolbar settings for the "Classic Web Version" mode and which buttons are available in the application when accessed from a **Desktop** environment.

Mobile

Here we must inform the toolbar settings for the "Mobile Version" mode. That is which buttons are available in the application when accessed from a **Mobile** dispositive.



Navigation:

Buttons relative to the navigation of the application.

Next	Move to the next page that can be a single record or a list of records.
Previous	Returns to displays the previous page records or a single record.
First	Move to the First page or record
Last	Move to the Last page or record
Exit	Close the application
Navigation by page	Displays a “page-number” navigation bar. **Example:** 1 2 3 4 5
Reload	Displays a button to reload the query data

Export:

Groups the options relative to the generated exports. Scriptcase generates the following export formats:

PDF	Generates a complete Report with all the data of the application in a PDF format.
WORD	Generates a complete Report with all the data of the application in a WORD format.
XLS	Generates a complete Report with all the data of the application in an EXCEL format.
XML	Generates a complete Report with all the data of the application in an XML format.
CSV	Generates a complete Report with all the data of the application in a CSV format.
RTF	Generates a complete Report with all the data of the application in an RTF format.
Print	Creates an HTML with the records ready for printing.

Export by Email:

Groups the options relative to the emails exports. Scriptcase generates the following export formats:

PDF (Email)	Sends by Email a complete report with all the data of the application in a PDF format.
WORD (Email)	Sends by Email a complete report with all the data of the application in a WORD format.
XLS (Email)	Sends by Email a complete report with all the data of the application in an EXCEL format.
XML (Email)	Sends by Email a complete report with all the data of the application in an XML format.
CSV (Email)	Sends by Email a complete report with all the data of the application in a CSV format.
RTF (Email)	Sends by Email a complete report with all the data of the application in an RTF format.

Others:

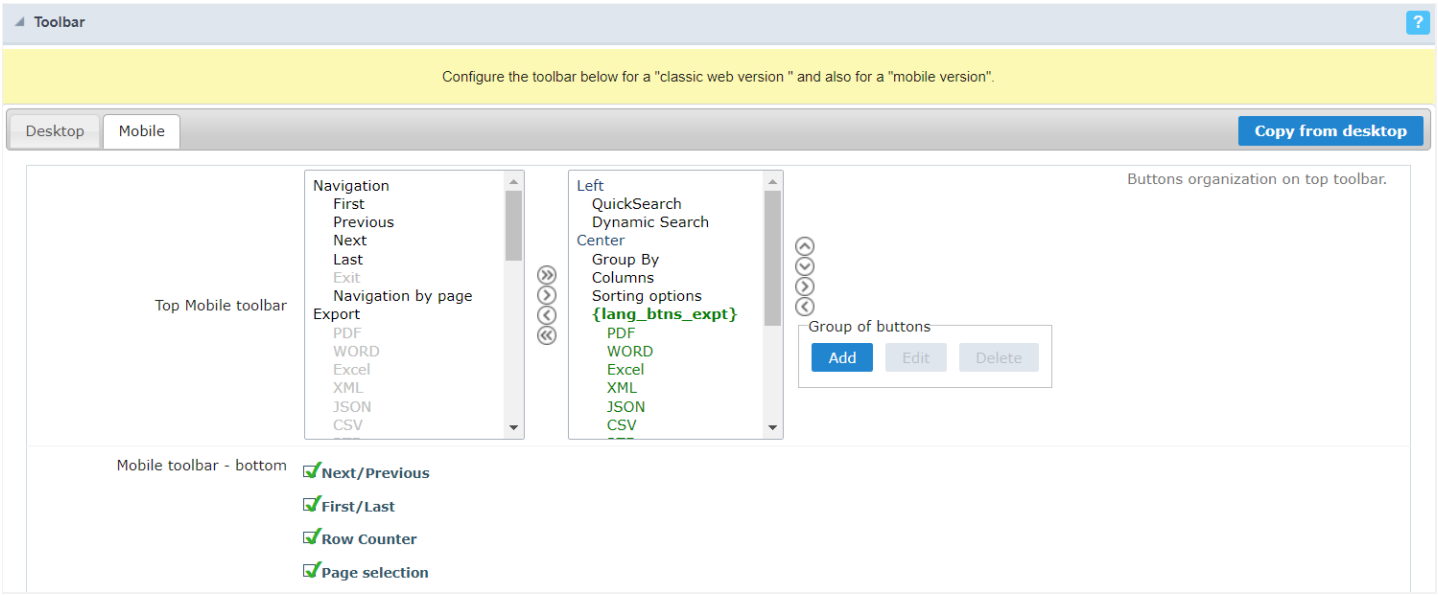
Other options available in the Grid application.

Jump to	Move to the informed page.
Rows Limit	It is a Combobox that defines the number of rows per page.
Search	Goes to the Search Form to filter the records.
Dynamic Search	It displays the fields of the search to filter the records.
Columns	Allows to include or remove columns of the Grid on the fly.
Sorting Options	Allows selecting the order of the records based on the field.
Group By	Allows to select or change a Group By rule on the Grid.
Save Grid	Allows saving the current state of the application. For example, in the advanced search, you can save the search data for further use.
Quick Search	Allows to perform a quick search in the records of the application.
Gantt	It displays a Gantt chart, if it was previously set.
Summary	It displays a summary with the synthetical data of the records.
Form Buttons	It displays the Form Buttons when there's an Application Link from the Grid to a Form.
Languages	It displays a Combobox with the names available, defined in the project properties.
Themes	It displays a Combobox with the themes available, defined in the project properties.
Rows Counter	It displays the number of records retrieved by the application.
HelpCase	Displays a button to open the help page.

Separator:

————————	Displays a line separating the buttons, when used the Group Buttons.

Toolbar Mobile



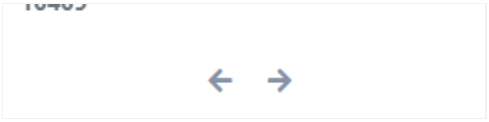
Top Mobile toolbar

It has the same options as the **Desktop** version, adding only the item “Copy from desktop”, which, when clicked, makes a copy of the items from the upper toolbar of **Desktop** to **Mobile**.

Mobile toolbar - bottom

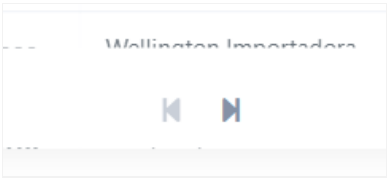
Next/Previous

Enables navigation to the next and previous page on mobile devices.



First/Last

Enables first and last page navigation on mobile devices.

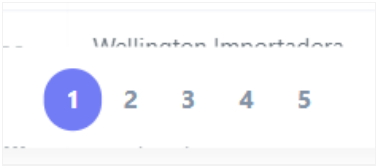


Row Counter

Enables the record counter showing the application’s total records

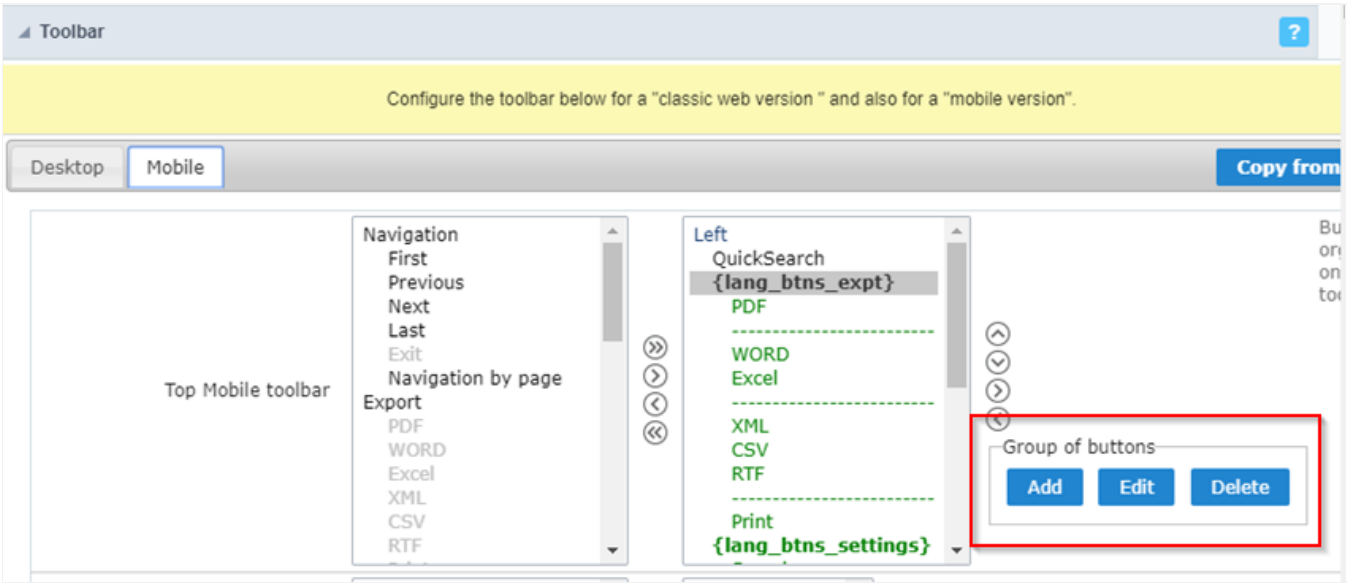
Page selection

Enables page navigation on mobile.



Bottons Group

The **Group** option allows you to group a set of buttons of the application toolbar to display them as a dropdown, for example.



Add

Add a new group of buttons.

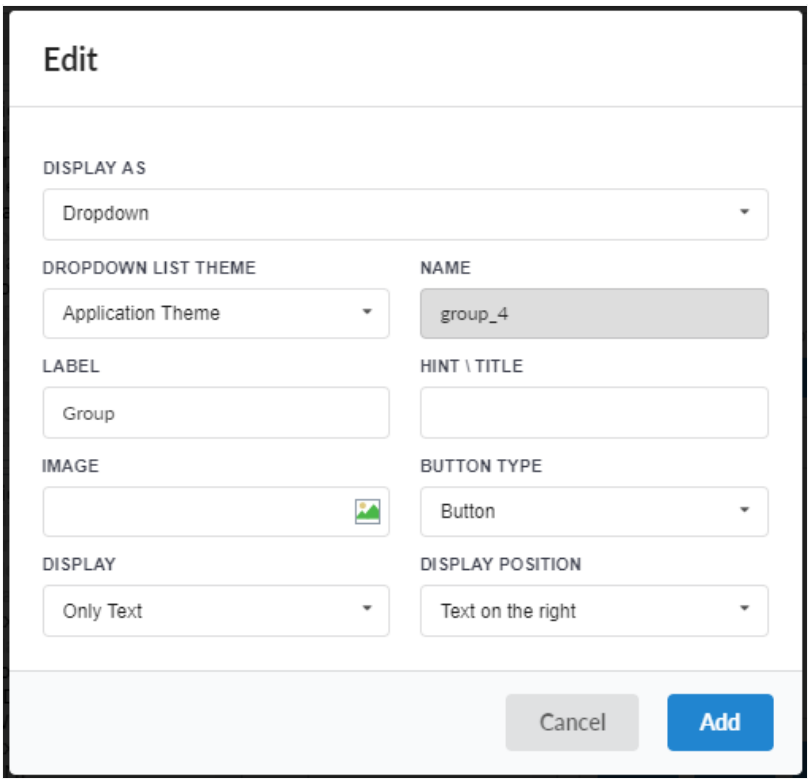
Edit

Edit an existent group of buttons.

Delete

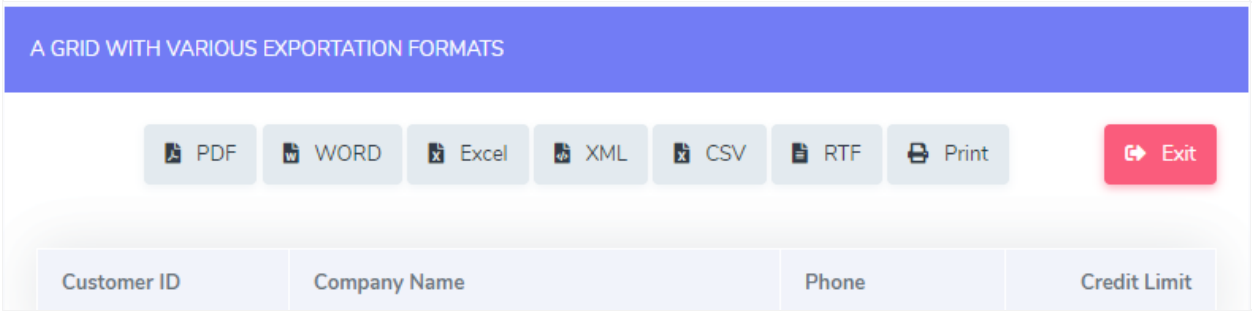
Delete the selected group of buttons.

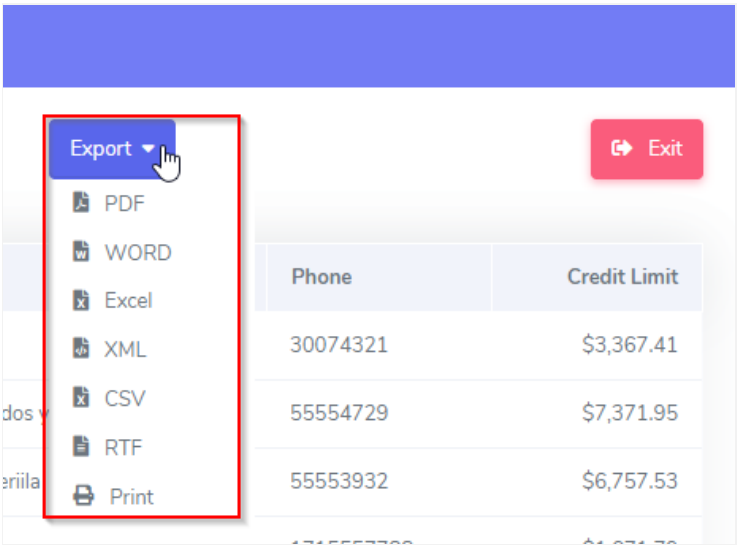
When you press the **Add** or **Edit** option, you can see the settings to configure the grouper:

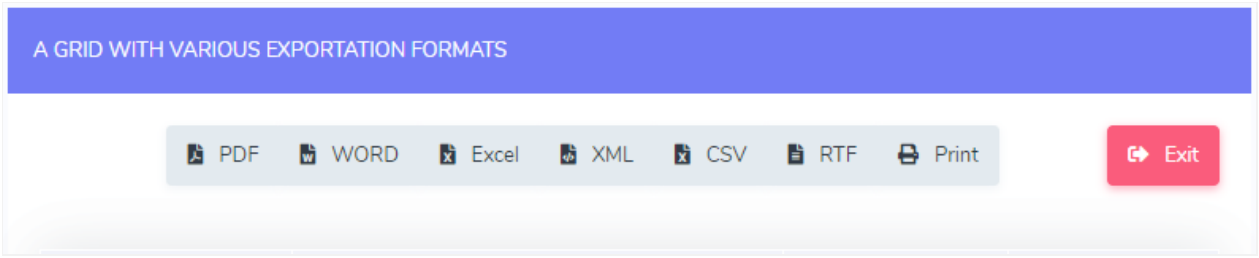


Display As

Allows displaying the group button as **Dropdown** or **Side by Side**.







Dropdown List Theme

Allows defining the Dropdown theme selecting between **Application theme** and **Button theme**.

Name

Allows defining a name for the button group.

Label

It is the displayed name for the button group in the application.

Hint\Title

Displays a hint to the end-user when the mouse is on the group of buttons.

Button Type

Allows displaying the button group as a Button, Image, or Link.

Image

Allows selecting an image for the button.

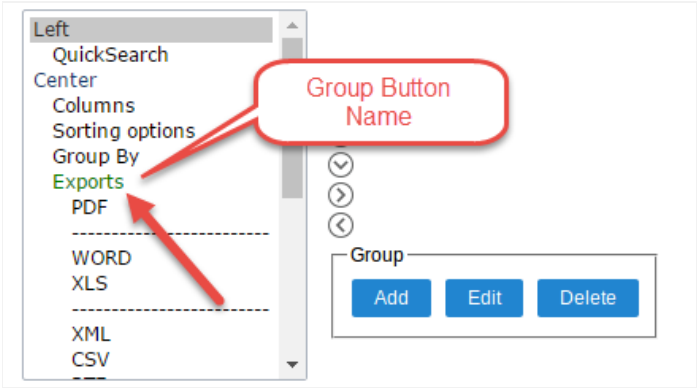
Display

Defines if the button displays only Text, only image, or both.

Display Position

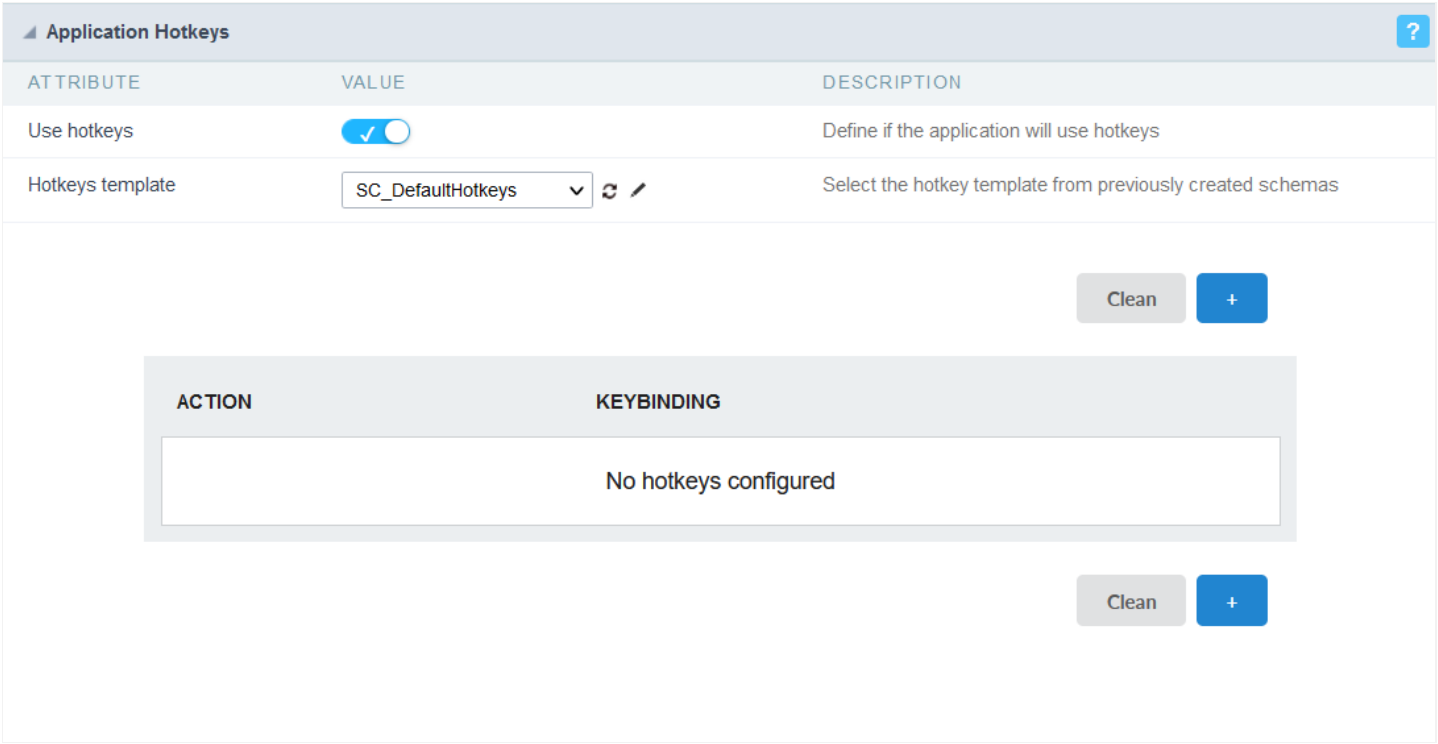
Defines the position of the Text or Image (Text to the right, Image to the right).

After creating a button group, you need to move the grouped buttons below of the Button Group and then move them to the right. Like the image below:



Application Hotkeys

Scriptcase allows creating shortcut keys to your applications. You can select a predefined template or create specific actions for an application.



Use hotkeys

Defines if the application uses hotkeys. When you enable this option, the default shortcut keys settings are disabled.

Hotkeys templat

Select the hotkey template previously created.

Action

Selects the triggered action when pressing the selected key.

Keybinding

Selects the keys responsible for executing the chosen action.

Add “+”

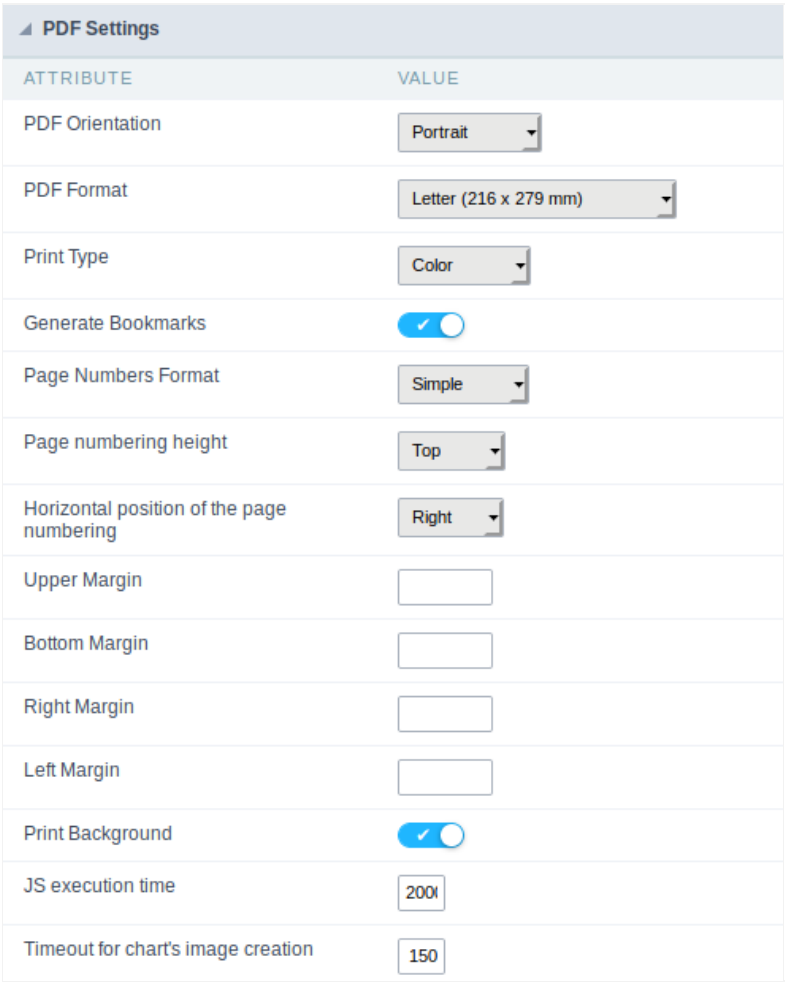
Adds a new action on the keys list.

Clear

It clears the selected hotkeys preference.

Chart Export Settings

PDF Settings



PDF Orientation

Allows you to set whether to print in Portrait or Landscape orientation.

PDF Format

It allows you to define the PDF paper format (letter, A4, others).

Print Type

It allows you to set whether the print mode as colored or economical with no colors.

Generate PDF directly

Opens the generated the PDF without the need to display an intermediate page with a link to download the file.

PDF Chart depth output selection

It allows the end-user to decide on the fly the depth of the Group By to exports.

Selecting page break per level

It allows the end-user to define the break level to skip to the next page in the generated PDF.

Configurable PDF

It allows the end-user to configure the PDF parameters on the fly.

Generate bookmarks

Generate bookmarks automatically according to the Group By.

Page numbering format

Set the numbering format as Simple (1,2,3,4,5, ...), Full (1 / n, 2 / n, 3 / n, ...) or no numbering.

Page numbering height

It allows you to align the numbering vertically, at the top or bottom of the page.

Horizontal position of page numbering

It allows the horizontal alignment of page numbering (Left, Center, Right).

Print background

Allows you to print background in the PDF file.

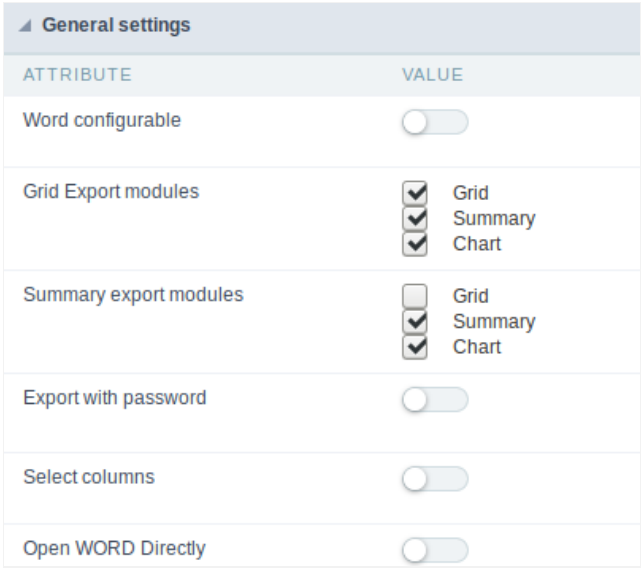
JS Runtime

The maximum time, in seconds, of server waiting when running JS.

Timeout for chart's image creation

Set the timeout to wait for the chart image creation in the PDF

WORD Settings



Word configurable

It allows the end-user to configure the export parameters on the fly.

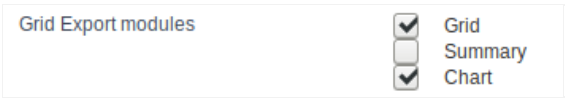
Export Modules of Grid

This option allows the developer to choose which modules are available in the grid export. The user can select the modules at

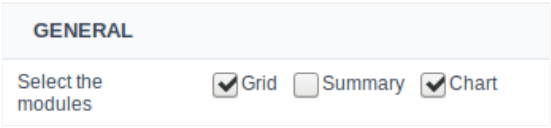
runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

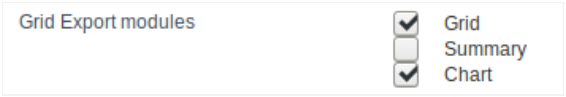


Summary Export Modules

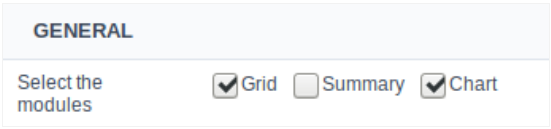
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.



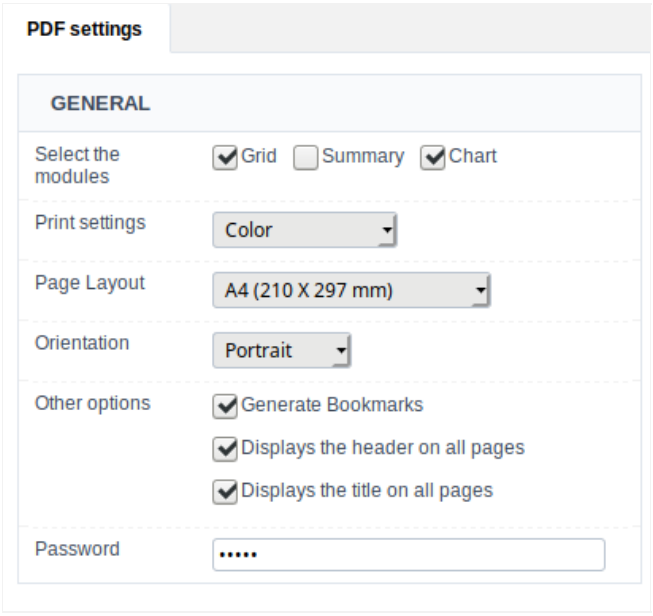
Export with password

This option allows the developer to set a password to export the file.

Password

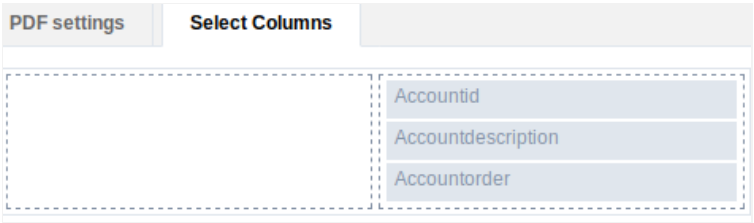
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

It allows the user has the option to choose the columns he wants to export at runtime.



Open WORD Directly

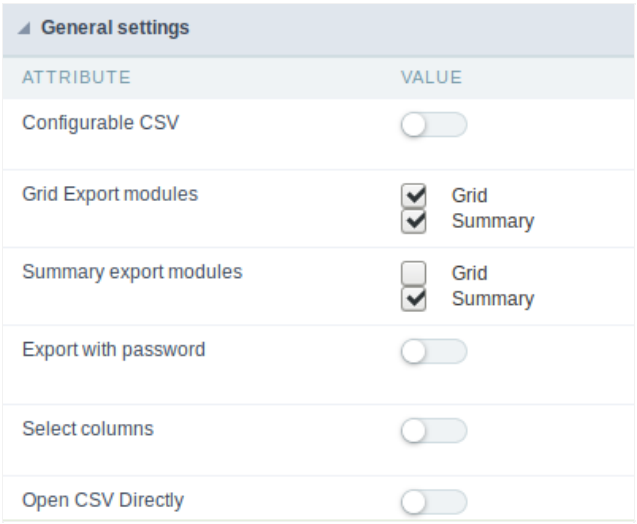
Opens the generated WORD file without the need to display an intermediate page with a link to download it.

WORD Settings

Print Type

It allows you to set whether the print mode as colored or economical with no colors(Black and White).

CSV Settings



Configurable CSV

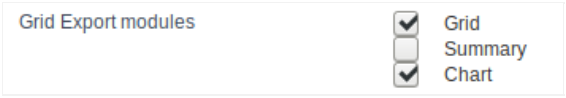
It allows the end-user to configure the CSV export parameters on the fly.

Export Modules of Grid

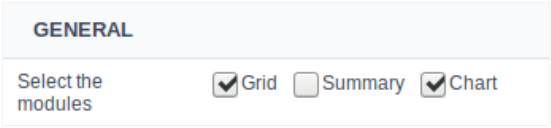
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

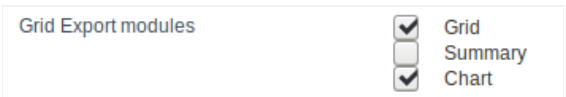


Summary Export Modules

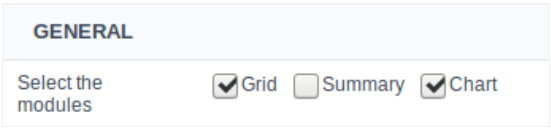
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.

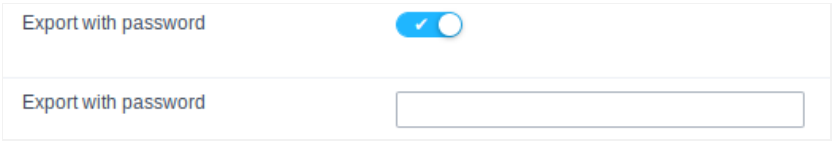


You can see the Summary option disabled at running time.



Export with password

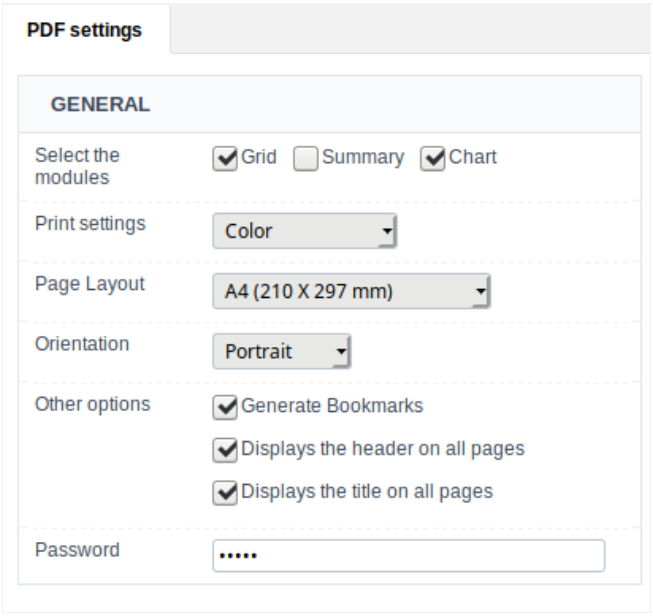
This option allows the developer to set a password to export the file.



Password

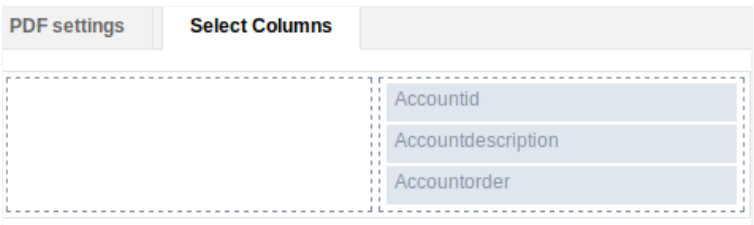
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

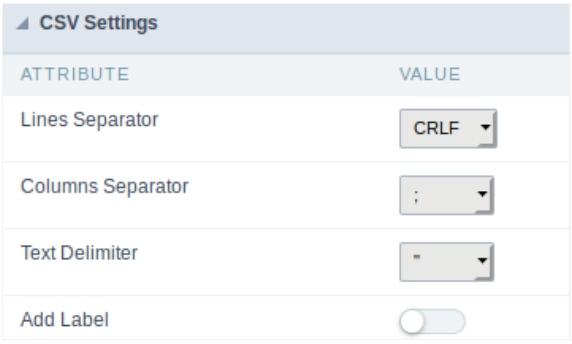
It allows the user has the option to choose the columns he wants to export at runtime.



Open CSV Directly

Opens the generated CSV file without the need to display an intermediate page with a link to download it.

CSV Settings



Line separator

It allows you to define the line separator character, each line representing a record.

Column separator

It allows you to define the column separator character, each column representing a database field.

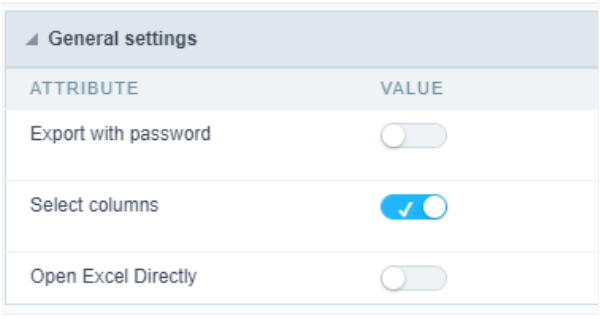
Text delimiter

It allows you to define the character used to delimit the text of the columns when we have database fields of type String.

Add label

It allows you to define displaying the label of the columns in the first line of the file.

XLS Settings



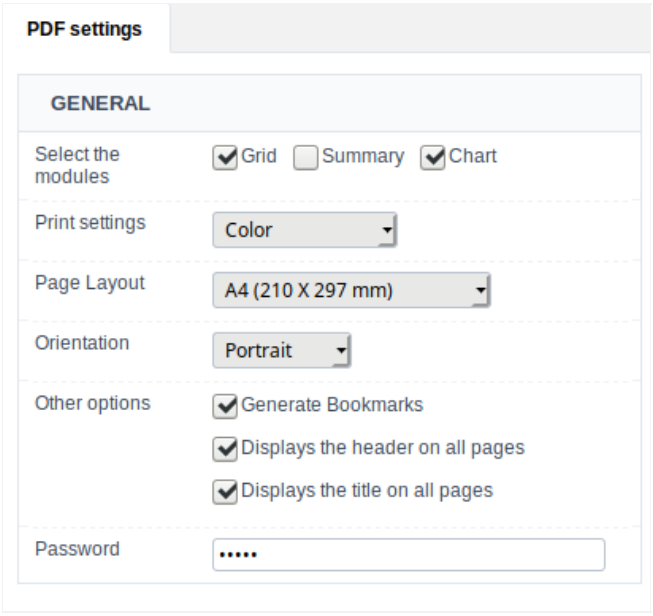
Export with password

This option allows the developer to set a password to export the file.

Password

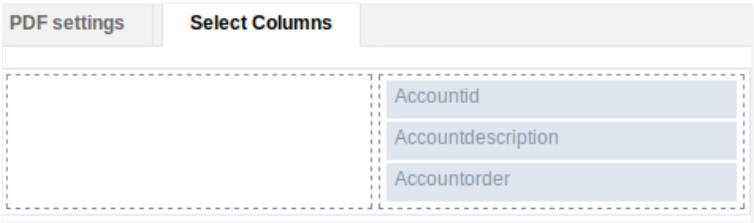
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

It allows the user has the option to choose the columns he wants to export at runtime.



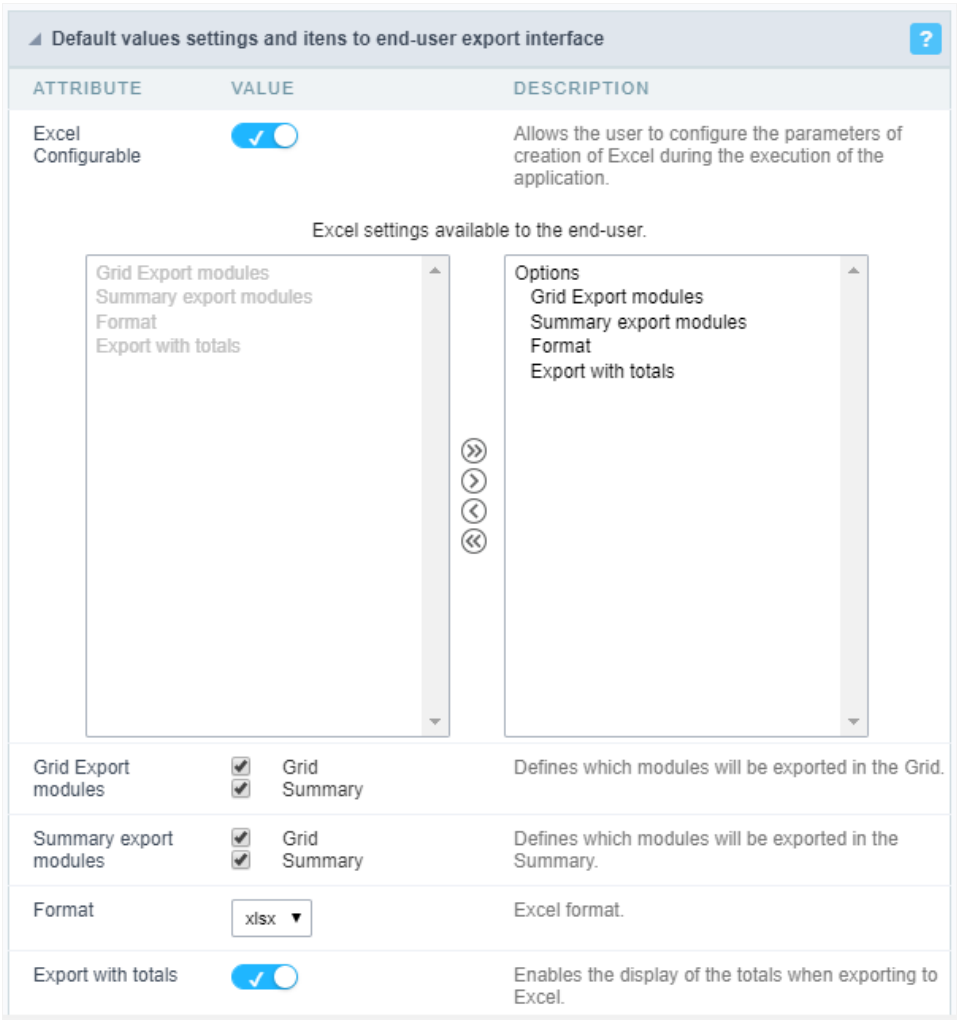
XLS Configurable

It allows the end-user to configure the XLS export parameters on the fly.

Open XLS Directly

It allows the end-user to configure the XLS export parameters on the fly.

XLS Settings

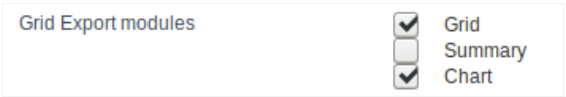


Export Modules of Grid

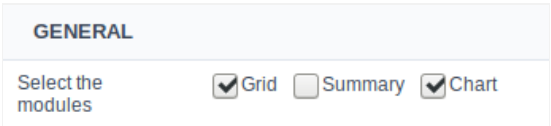
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

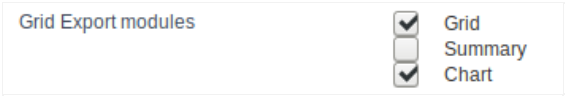


Summary Export Modules

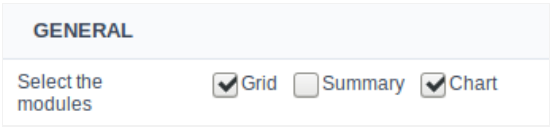
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.



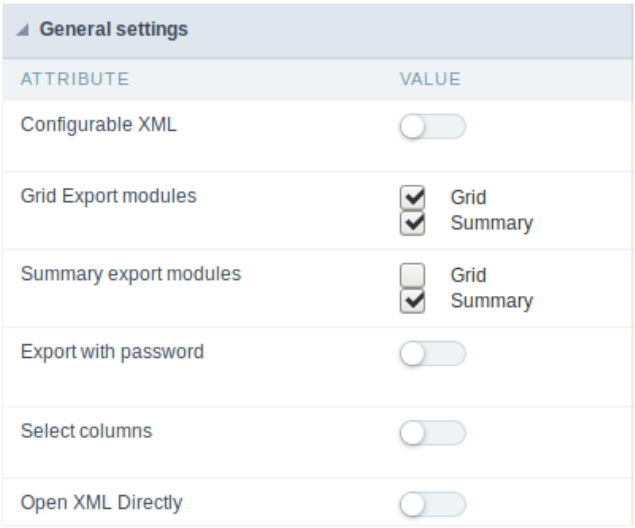
Format

It allows defining the Excel format of the generated document (xls or xlsx).

Export with totals

Show the totals when exporting the application data to Excel.

XML Settings



Configurable XML

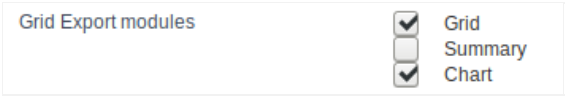
It allows the end-user to configure the XML export parameters on the fly.

Export Modules of Grid

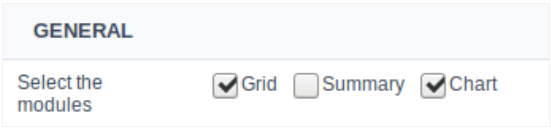
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

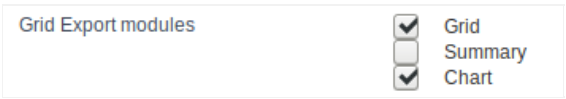


Summary Export Modules

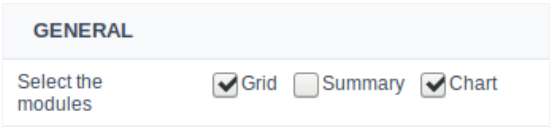
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.



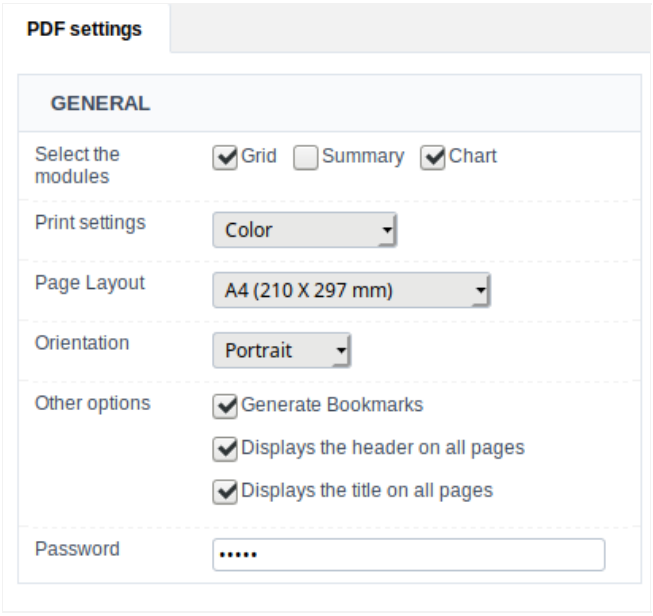
Export with password

This option allows the developer to set a password to export the file.

Password

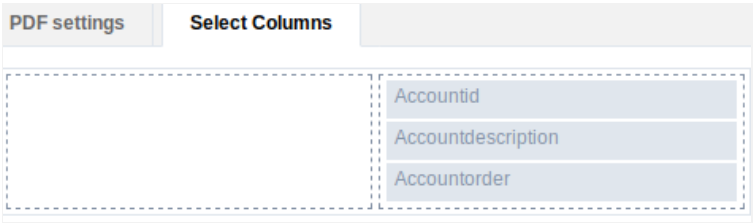
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

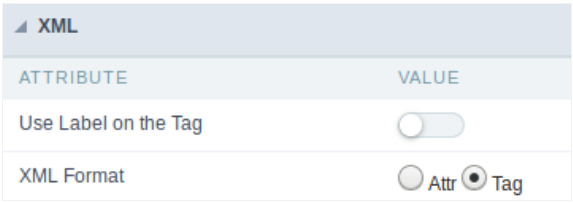
It allows the user has the option to choose the columns he wants to export at runtime.



Open XML Directly

Opens the generated XML file without the need to display an intermediate page with a link to download it.

XML Settings



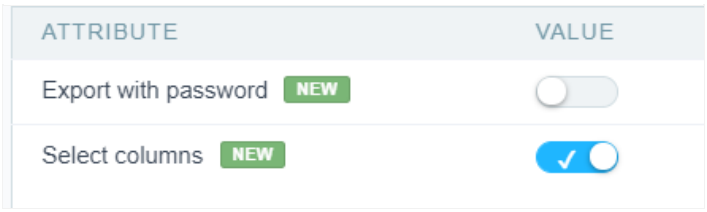
Use Label on the Tag

It allows using the field labels on Tags of the XML file.

XML Format

Define how to generate the records in the XML file, if it will store the values on attributes or new elements. (Attr or Tag)

JSON Settings



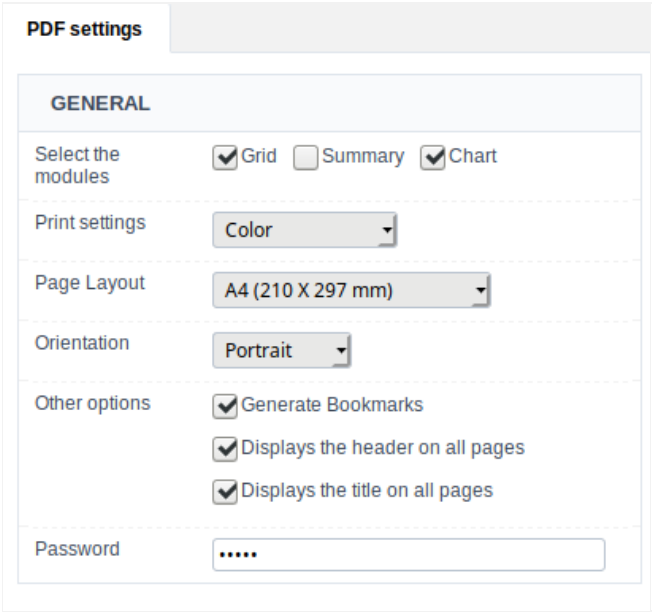
Export with password

This option allows the developer to set a password to export the file.

Password

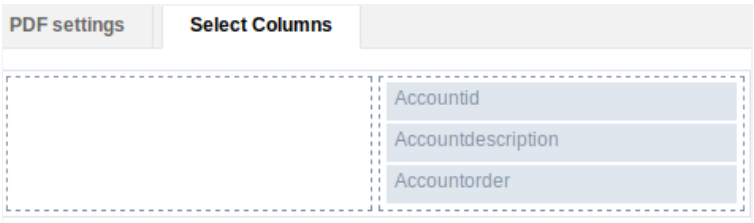
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

It allows the user has the option to choose the columns he wants to export at runtime.



Open JSON Directly

It allows the end-user to configure the JSON export parameters on the fly.

JSON Settings

Configurable JSON

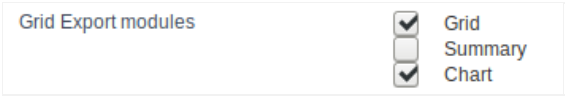
It allows the user to configure the XML creation parameters during the execution of the application. If this option is disabled, the end user will only have access to the settings defined by the developer.

Export Modules of Grid

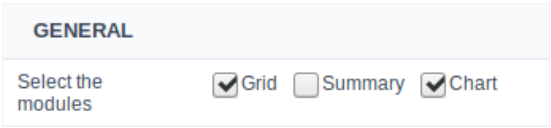
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

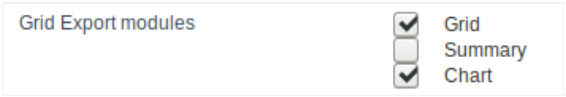


Summary Export Modules

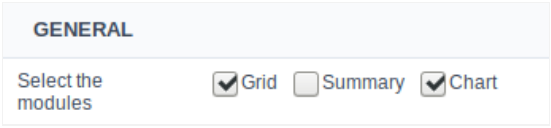
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.



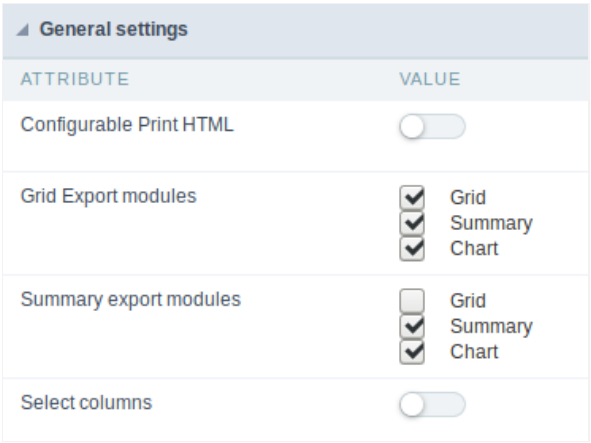
Use Label

It allows the label used in the application field to be sent in the JSON attribute.

Formatted value

It allows displaying the formatted numeric value or the value displayed in the database.

Print Settings



Configurable Print HTML

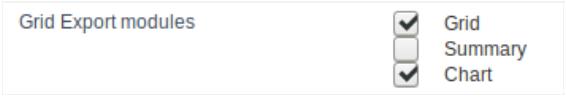
It allows the end-user to configure the HTML Printing parameters on the fly.

Export Modules of Grid

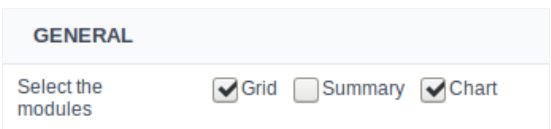
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

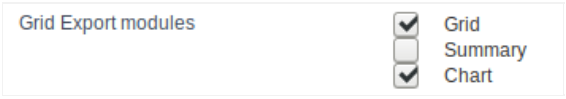


Summary Export Modules

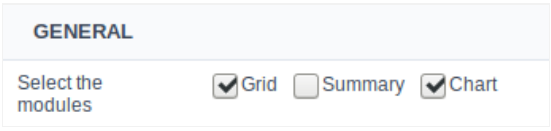
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.

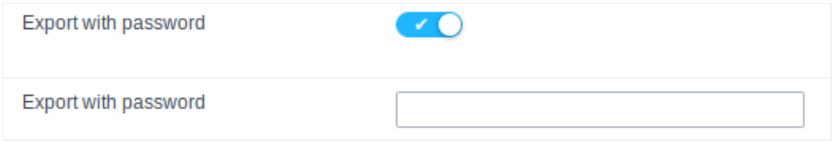


You can see the Summary option disabled at running time.



Export with password

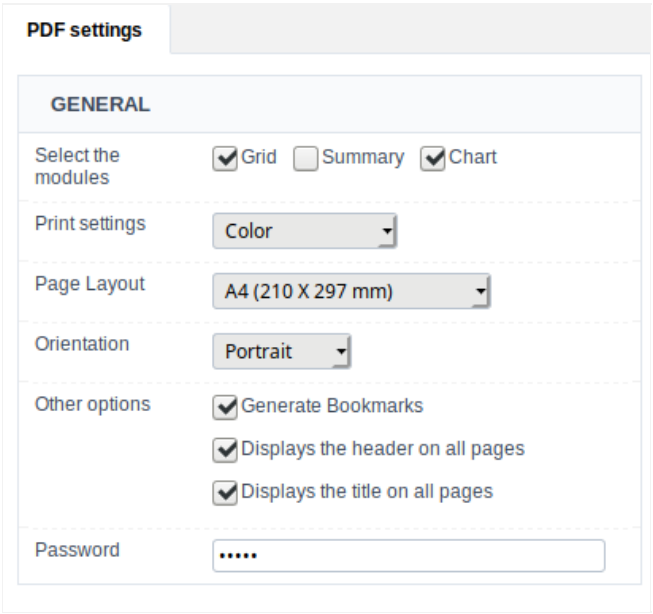
This option allows the developer to set a password to export the file.



Password

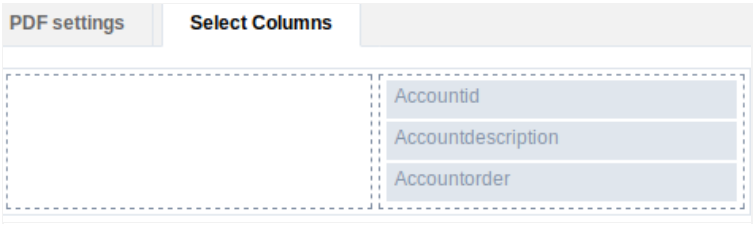
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

It allows the user has the option to choose the columns he wants to export at runtime.



Print Mode

It allows choosing the content to print. (Current Page or Full Report)

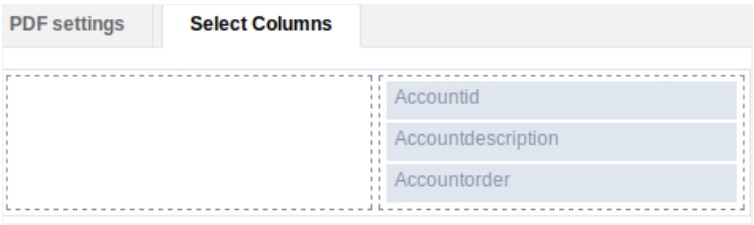
Print Background

Allows displaying the background in an HTML form.

RTF Settings

Select columns

This option allows the end-user to select which fields he wants to export to the file.



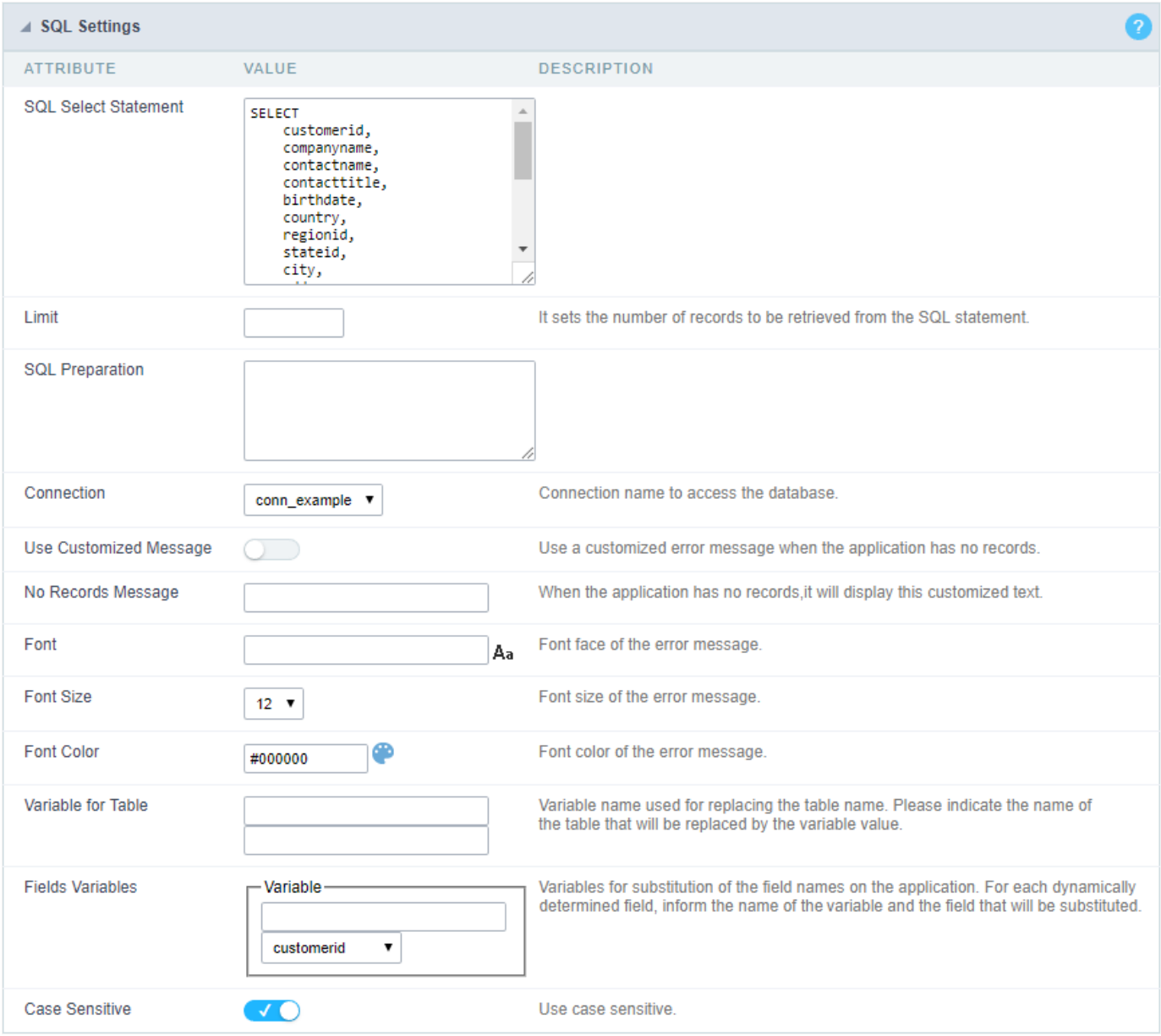
Open RTF Directly

Opens the generated RTF file without the need to display an intermediate page with a link to download it.

Chart SQL Settings

SQL Settings

This interface allows configuring the related database settings, such as the SQL statement, the used database connection, case sensitive, and others.



SQL configuration

SQL Select Statement

It allows you to define the primary SQL of the application. You can edit this SQL to add or delete fields.

Limit

Lets you limit the display in the number of records retrieved by SQL query.

SQL Preparation

You can enter SQL commands or procedure names to execute them before the primary SQL of the application.

Grid

Connection

It allows defining the database connection of the application. You can change the connection to another one that has the same table.

Use Customized Message

Lets you define to display the “no records” message or not.

No Records Message

Lets you set the message when the application has no records.

Font

This option is available when using the “Use Customized Message” option. It lets you set the font for the message.

Font Size

This option is available when using the “Use Customized Message” option. It lets you to set the font size.

Font Color

This option is available when using the “Use Customized Message” option. It lets you to set the font color.

Variable for Table

It allows to use a variable to change a part of the string containing the table name.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the table you want to replace (replace from).

Fields Variables

It allows to use a variable to change a part of the string containing the field name.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the field you want to replace (replace from).

Case sensitive

It defines if the database connection uses case sensitive or not.

Chart Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application.

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Chart Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Chart Headers

Desabilitar o Auditor XSS

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

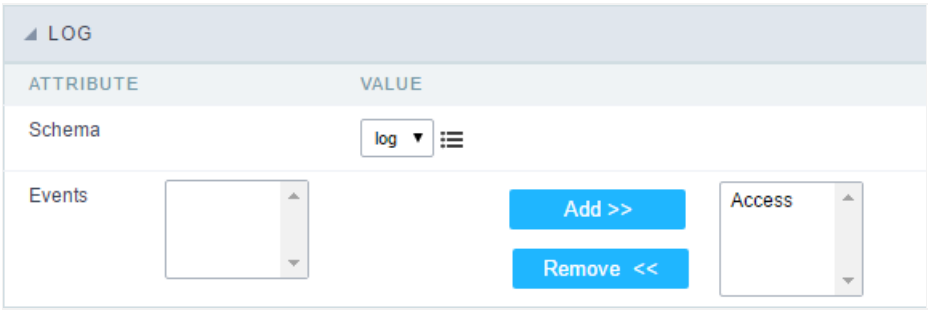
Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Chart Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Chats - Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid		
Procedure		
Chart		
Form		
Control		
Calendar		
Search		
PDF Report		

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven't manage fields.

Chart - Text Field

General Settings

This type of field allows the developer to create quickly inputs to insert and update data, where the final user can inform its data to be allocated in its database.

Type of Data

Defines the type of the field in the application. In this case we should select **Text**.

Label

Defines the title that will be displayed in the field when the application is executed. The terminology used in the interface is fundamental to the system have a good usability, whe should use names and familiar terms to the final user of the application, instead of using terms from the system.

For example, this text field that has the name **customerid**, the client would have a much better understanding of the functionality of the field when we define the label as **Customer Name**.

Not only a fixed text, the **Label** attribute allows the use of langs to define the field's title, making it possible to use your application in a multi language project.

SQL Type

Informs the type of the field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  ```
  SELECT display_field FROM table WHERE key_field = {grid_field}
  ```

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select

command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the

select command separated by the character in the “Separated by” field.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Chart - Multiple Lines Text Field

General Settings

Multiple Lines Text field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Multiple Lines Text , you can inform a Text value to the field in multiple lines.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **SQL Type** : Informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  ```
  SELECT display_field FROM table WHERE key_field = {grid_field}
  ```

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Chart - Integer Field

General Settings

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Label:

It sets the title that will be displayed in the field when you run the application. The terminology used in the interface is fundamental for your system has a good usability. In this case, we recommend you to use names and terms familiar to the end user of the application, instead of using terms originated in the system.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Repeat value:

This option when enabled will allows you to repeat the field value if it is equal to the value of the previous record in the database.

Example:

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:

Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Chart - Decimal Field

General Settings

Decimal field configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

SQL Type

It informs the data type of field in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Chart - Percent Field

General Settings

Percentage field configuration Interface.

Data Type

Define the type of field to Percentage.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

SQL Type

It informs the data type of field in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Chart - Currency Field

General Settings

Currency field configuration Interface.

Data Type

Define the field type to Currency. It allows defining the currency number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

SQL Type

It informs the data type of field in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Chart - Percent(Calculated) Field

General Settings

Percentage field configuration Interface.

Data Type

Define the type of field to Percentage.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

SQL Type

It informs the data type of field in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Chart - Date Field

General Settings

Date field Configuration Interface.

Data Type

Define the type of field. When setting it to Date, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Chart - Time Field

General Settings

Time field Configuration Interface.

Data Type

Define the type of field. When setting it to Time, you can inform a time format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

SQL Type

It informs the data type of field in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Chart - Date and Time Field

General Settings

Date and Time field Configuration Interface.

Data Type

Define the type of field. When setting it to Date and Time, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Date format

It formats the date for the Group By.

SQL Type

It informs the data type of field in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

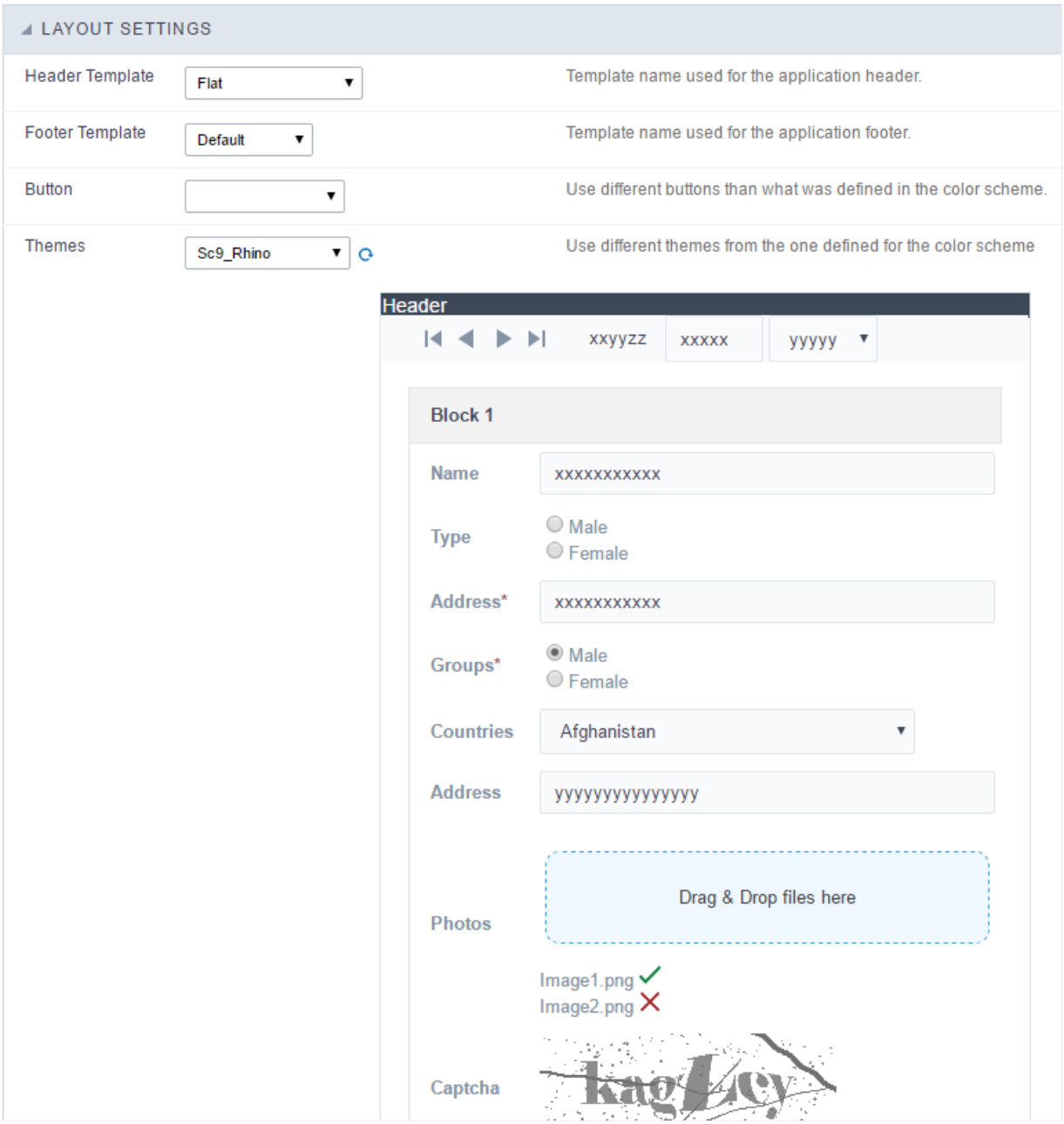
Sets the maximum range of the Spin.

Chart Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout Settings

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

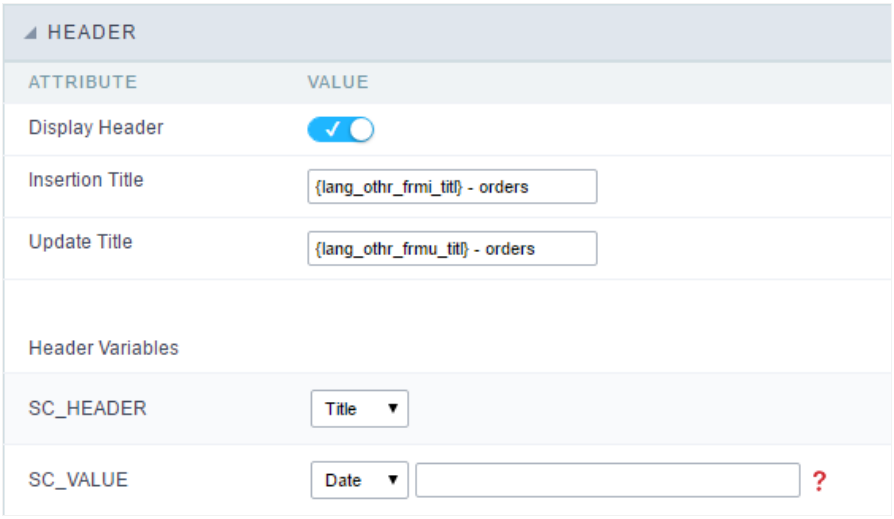
Allows choosing the button theme for the application.

Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

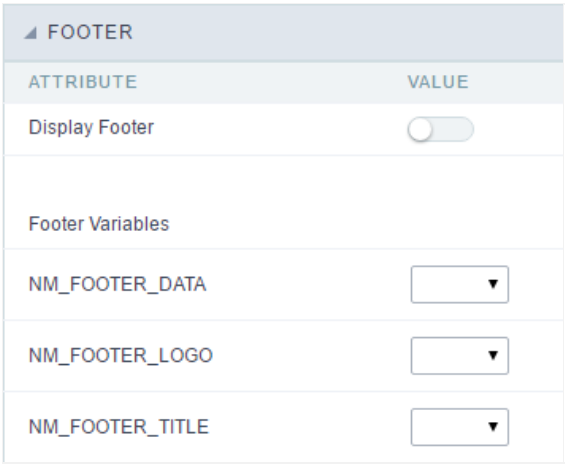
Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Chart´ Events Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application’s CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event’s PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Chart

It only occurs one time when the application loads. Before the application run SQL. It can be used for data processing or for checking variables. Ex: if ([glo_var_department] != 'financial'){ sc_redir(app_x.php); }

onScriptInit - Chart

This event occurs when the application its loaded, or reload. Occurs before the application runs the SQL statement, so using this event it is possible to modify the grid SQL statement dynamically, based on any logical. Macros frequently called on this event: sc_select_field, sc_select_order, sc_select_where(add), etc

onHeader - Chart

This event is frequently used when it is required to display any calculated value on report's header Ex. In a customer report header it is required to display the customer balance on reports header. Create a field called fld_customer_balance on fields menu. sc_lookup(v_bal," select sum(debits) - sum(credits) from view_debits_credits where customer ='{field_customer}' ");

onFooter - Chart

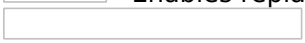
This event occurs on the application footer display, it is frequently used when it is required to display any calculated data on grids or report footer. Ex: sc_lookup(ret," select sum({item_value}) from orders_items where id_order='{fld_id_order}' "); {fld_footer_total_field_a} = {ret[0][0]} * 0,1; The field {fld_footer_total_field_a} must be created on fields menu

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

> Click Here to view the Scriptcase hotkeys documentation.

Chart Search Settings

With this interface, you can define general options of the Search Form.

Search configuration Interface.

Search Criteria

Allows to select the logical operator **AND** or **OR** to define the criteria of the search;

Display Condition

Gets the condition of the search available for the user to choose one. He can select “AND” or “OR” in a Combobox.

Use auto-complete in the fields

Automatically turns the field into an autocomplete according to the existing values in the database. If the user chooses **Yes**, the autocomplete will enable automatically in all inputs that contain a relationship. If the user decides **No**, so no autocompletes will be displayed. Otherwise, the option selected is **Defined in the field** it’ll keep the settings for each field individually.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Ads
- Search with Ads Advanced and Dynamic search
- Pivot Table with Filter
- Macro to Write Filter
- Highlight option

Related Videos

Chart Search Criteria Settings

With this interface, you can configure the conditions available for each field of the Search form.

Search configuration Interface.

We can see the fields list on the left combo. On the right, the list of options for filtering the selected field. To select an option, click on one of them (Equal to, Beginning with, Contains, etc.) and then the button On/Off. The arrows, on the right, allows altering the order of the fields.

For the Date type fields, you can define special conditions for the search, accessing the field configurations, and editing the Special Conditions Settings.

Below the list are the buttons to enable the selected options:

- **On/Off**: Enables or disables the field or the option chosen.
- **All**: Marks all fields or options.
- **None**: Unmarks all the fields or options.

Related Links

- Advanced search with a predefined initial value

Related Videos

- Dynamic Search
- Advanced search with relative periods
- Refine search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Chart Advanced Search Settings

Through the table below we set all the options that will be part of the application Grid Search.

Margins

Defines the position of the margins of the Search Form.

Keep Values

It keeps the searched values when the user returns to the search form.

Keep Columns and Order Selection

Set it to preserve the selected columns and sorting for each search, if they went changed by the user through the toolbar options.

Use Enter to

It allows you to define the action that the Enter Key has on the Search form. **Tabulate** enables you to navigate between fields, and **Submit** performs the search(activates the Search button).

Display Tags

Allows displaying as tags, the searches used for the Grid.

Display after filtering

Display tags only after performing an advanced search. If disabled, it will always display a tag, regardless of the advanced search.

Unify results

Sets the chars limit to group the result of the tags. This option should be used when the field type is multiple-select.

Treeview in the Tags

Sets the use of Treeview for tags.

Initial status of the Treeview

It sets the initial state of the Treeview. (If the app is using Treeview for tags)

Start open

It displays the tags.

Start close

It displays the full description of the tags as text.

Clear other filters after submit

After applying the advanced filter, all other filters will be deleted.

Interact with the dynamic filter

Apply the same filter in matching fields between Advanced and Dynamic Filter, when not showing the labels.

Highlight

Highlights the results in the query. It only works for “Exactly the same”, “Contains” and “Equal start” conditions.

To customize how the highlight will be displayed, you must access the theme editing tool .

1 - Access the **Layout > Application Themes** menu.
2 - Choose the desired theme and click **Advanced Mode**.
3 - Then look for the sub-item **Grid**, within the item of the same name.
4 - In this sub-item you will find the option “Line”, which contains the folders **Line Odd and Even** . In each of these folders you will find the option **Quicksearch highlight**, with the options for editing Text, Border and Background of the searched terms.

Related Links

- Search with buttons in line
- Search with a predefined initial value
- Search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos

Select Fields for Advanced Search in Chart

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

- - Move all fields to the right.
- - Moves only selected fields to the right
- - Moves only the selected fields to the left.
- - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the button.

> To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning

Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

> Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro se_where_filter

Related Videos

Imagens utilizadas no arquivo includes/docs/app/posicionamento_campos –> [img_posicionamento_campos]: /assets/images/docs/app/comum/campos/posicionamento_campos/posicionamento_campos_search.png

Required Fields For Advanced Search in Chart

Defines which fields of application will be required for the search.

The application generated will be displayed a bullet (*) next to the field and an error message is generated if not assigned no value.

- **Marker position** : Marker’s position relative to the field.
- **Display message** : Displays whether or not the validation error message.

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos

Chart Search Toolbar

Search Toolbar

The Search toolbar is divided in two parts: Top and Bottom, in a way that is possible to define the buttons that will be displayed in both bars. The selection of buttons in the top and bottom toolbar works independently, allowing the buttons to be displayed in both bars at the same time.

Toolbar Interface.

Navigation:

Groups the options relative to the navigation buttons that can be displayed in the application.

- ***Search***: Execute the search.
- ***Clean***: Clean the all the search fields.
- ***Edit***: Enable the ***Save Tag*** option.
- ***Exit***: Exit the application.

Others:

Groups a diversity of options relative to the application.

- ***Languages***: Displays a combobox with the names available, defined in the project properties.
- ***Themes***: Displays a combobox with the themes available, defined in the project properties.
- ***HelpCase***: Displays a button to redirect to the help page.

Separator

- **———————**: Dispays a line separating the buttons, when used the Group Buttons.

Use in-line buttons:

Allows the alignment of the filter buttons next to the fields.

Inline buttons: Allows you to select which buttons will be displayed next to the field, and you can sort them according to your wishes. This option is available by enabling **Use buttons inline** in the button settings.

When activating the **Inline Buttons** option, the screen to configure the buttons will be displayed.

> The buttons available in this in-line button option are the same as those shown in the standard toolbar described above.

This is the result when using a radio button on the line

Button Settings

Button Settings Interface.

- **Hotkey**: Allows you to set keyboard hotkeys to a button.
- **Position of the in-line buttons**: Sets the positioning of the buttons to the right or left of the fields.
- **Column Quantity**: Sets the number of display columns of the buttons, allowing you to configure whether they will be displayed side-by-side or distributed in columns.

Options ![Options Interface.][barra_ferramentas_filtro_opcoes] *Options Interface.* * __Button Position(Top/Bottom)__ Positioning the buttons of the toolbar Top/Bottom. -->

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value

Related Videos ▷

- [arch with relative periods](#)
- [Search with blocks](#)

- [Highlight option](#)
- [Macro sc_where_filter](#)

Save Search in Advanced Search (Chart)

This feature allows the end-user to save his searches in a profile. You can create some rules, like to save the searches by user login.

 Save Filter Interface.

Save Filter Interface.

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos

Grid’s Advanced Search Layout Overview

In events, blocks can be used with global variables, local, JavaScript code, CSS codes and Scriptcase macros.

onScripInit - Advanced Search - Chart

This event occurs just once before it display the search form. It can be used to set some default values to the fields. For example: {company} = [glo_company]; Date fields are used differently, They use a suffix "day" for day, "month" for month and/or "year" for year. For example, The {Birth} field will be: {Birth_day}=date("d"); {Birth_month}=date("m"); {Birth_year}=date("Y");

onRefresh - Chart

onRefresh

This event occurs when the form application is reloaded, is possible to reload the form based on fields (select , radio, checkbox)

onSave - Advanced Search - Chart

This event occurs always that some search setting is saved in the filter application.

onValidade

This event occours when the search form is submitted. Ex. in order to prevent some searches or to validate any search params if(strlen({name}) <= 0) { sc_error_message(" Please fill the name field before submit your search "); }

Chart’s Advanced Search Layout Overview

On this module are available the features of editing, attributes and display of the application, in a way that you can apply display themes, organize blocks, define values and the format of the Header/Footer and etc.

Blocks

Conceptually a block is a “container” where you can position the fields of the Applications: Form, Control or Grid with Slide orientation.

By default, the applications created in ScriptCase are built with a only one block, with the same name as the application. You can add the amount of blocks that you want to organize in a more convenient way. The page below, observe that theirs a column Organization, and that is where you’ll define if the next block will be set beside or below the current one.

Application Block configuration

On the left side of each block there are two icons, first has the function to edit all the information relative to the block and the second is to delete the block.

- Organizing the position of the Blocks
  - See below how to modify the display order of the Blocks in one Page.

  Click and drag the block that you desire to modify to its new position.

  Application Block Display configuration

  - See how to remove a block from display

  Click on the block desired and drag it to the item “Blocks not Shown”. This way, you can also drag the block to another page if desired. See the images below.

  Application Block Display configuration

  Application Block Display configuration

  - **Attributes**
    - **Block**
      - **Name** : Name of the Block.
      - **Label** : Title of the block that’ll be displayed in the application.
    - **Title**
      - **Display** : Flag that controls the title display of the block.
    - **Label**
      - **Display** : Flag that controls id the label of the fields will be displayed in the block.
      - **Position** : Options to display label :
        - **Above** : The label will be displayed above the field.
        - **Beside** : The label will be displayed beside the field.
        - **Below** : The label will be displayed below the field.
    - **Fields**
      - **Columns** : Amount of columns that are displayed side by side in the block.
      - **Position** : The way that the fields are displayed in the block :
        - **Below** : The fields are displayed one below the other respecting the amount of columns.
        - **Beside** : The fields are displayed one beside the other respecting the amount of columns.
        - **Line** : The fields are displayed one beside the other without the tabulation.
    - **Organization**
      - **Next** : The way that the blocks are displayed in the page:
        - **Below** : Indicates that the next block will be placed below the current one.
        - **Beside** : Indicates that the next block will be placed beside the current one.
        - **Tabs** : Indicates that the next block will be placed in a different tab then the current one.
      - **Width** : Specifies the width that block will occupy in pixels or percentage, in case the value is in percentage, inform the (%).
      - **Collapse** : Enables the option to close the block.

- Create New Block

  To include new blocks in an Application, click on the button . Next, you’ll see the following interface to define the name and label of the block. At the end click on Create.

Creating application blocks configuration

- **Attributes**
  - **Name** : Name of the Block.
  - **Label** : Title of the block that'll be displayed in the application.

- Edit Blocks

  To edit a block just click on the icon , that is on the left side of the block. Next, you'll see the following interface to define the parameters of the blocks. At the end click on save.

- **Name** : Name of the block.
- **Title** : Block title for display.
- **Display Title** : This option, when active, allows to display the block title.
- **Title Font** : Font applied to the block title.
- **Font Size** : Size of the font applied to the block title.
- **Font Color** : Font color for the block title.
- **Background Color** : Background Color of the block title.
- **Background image** : Background image for the block title.
- **Title Height** : Height in pixels of the block title line.
- **Horizontal Alignment** : Horizontal Alignment of the block title (Left, Center and Right).
- **Vertical Alignment** : Vertical Alignment the block title (Top, Middle and Bottom).
- **Display Label** : Display the labels of the fields in the block.
- **Columns** : Amount of field columns in a block.
- **Columns Width** : How the width of the block is defined.
- **Label Color** : Color of the field labels.
- **Fields Organization** : How the fields are organized in the block.
- **Label Position** : Position of the field labels relating to the data.
- **Next Block** : Position of the next block relating to the current block.
- **Border Color** : Border color for the block.
- **Border Width** : Border Width for the block.
- **Block Width** : Width for the block.
- **Block Height** : Height for the block.
- **Cell Spacing** : Cell Spacing in the block.
- **Collapse** : Enables the option to close the block.

Related Links

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos

Settings

On this interface, you can define the theme for display of a specific application, this being because ScriptCase uses a Standard Definition of Values per project, that besides the Display Theme, allows to define values for other attributes on a Project level.

Header Template:

Allows to choose the template that’s going to be used as the Header of the application.

Footer Template:

Allows to choose the template that’s going to be used as the Footer of the application.

Button:

Allows to choose the button theme for the application.

Themes:

Choose one of the existing themes, it’ll load the display mode (colors ,fonts, etc.) that’ll be part of the application.

Header & Footer

Header

In this block, it’s the definition of the variables content that’ll be part of the header.

This page may change depending on the header format chosen in the Layout Settings.

Display Header:

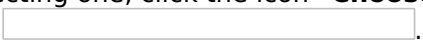
This option determines if the header will display.

Title:

Allows to inform the title displayed in the application.

Header Variables:

The field variables can be informed with anyone of displayed in the Combo box. Depending on the type, it’ll be necessary to associate the content with the field. Below there are the types of content:

- **Field** : When you choose the option “**Field**”, it’ll open a Combo box beside to choose the field you want. Selecting the desired field, it’ll associate the value of the field with the header.
- **Title** : This option when selected it’ll display in the header the value informed in the “**Application Title**”.
- **Date** : When selected the “**Date**” type, it’ll display the system’s date in the header. There are a diversity of formats using the date and time of the server. The format can be informed in the text field that appears beside the field. To access the existing formats, click on the icon and you’ll view a page display the formats.
- **Image** : When selecting the image type, it displays a field to inform the name of the existing image in the server. To locate the images existing and selecting one, click the icon “**Choose Image**” and to upload new images click on the button “**Upload**”  .
- **Value** : When selecting the type “**Value**”, the content informed in the text field that appears beside, it’ll be displayed in the header, you can inform texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

  Depending on the Application, you may have more than one title option.

Footer

This page may change depending on the footer format chosen in the Layout Settings.

Display Footer:

This option determines if the footer will display.

Footer Variables

The field variables can be informed with anyone of displayed in the Combo box. Depending on the type, it’ll be necessary to associate the content with the field. Below there are the types of content:

- **Field** : When you choose the option “**Field**”, it’ll open a Combo box beside to choose the field you want. Selecting the desired field, it’ll associate the value of the field with the footer.
- **Date** : When selected the “**Date**” type, it’ll display the system’s date in the footer. There are a diversity of formats using the date and time of the server. The format can be informed in the text field that appears beside the field. To access the existing formats, click on the icon and you’ll view a page display the formats.
- **Image** : When selecting the image type, it displays a field to inform the name of the existing image in the server. To locate the images existing and selecting one, click the icon “**Choose Image**” and to upload new images click on the button “**Upload**” .
- **Value** : When selecting the type “**Value**”, the content informed in the text field that appears beside,

it’ll be displayed in the footer, you can inform texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

Chart Search Settings

On this screen, the general behavior of the dynamic filter in the application will be defined.

Filter type

Defines the type of filter that will be used in the application.

Filter Builder

Allows the end user to create filters during the use of the application, combining different conditions and logical operators (AND/OR) to create more robust filters and generate more qualified reports.

Below is an example of how the filter builder works.

The user can add multiple fields or nested conditions to combine the use of logical operators.

Dynamic filter

Allows the end user to build filters during the use of the application by adding the fields that should compose the filter according to their needs, within the same logical operator.

Below is an example of how the dynamic filter works.

The user must select the logical operator that will be used for all filter fields: All conditions (AND) or any condition (OR).

Use the ENTER key for

This attribute defines the behavior of the ENTER key when the filter screen is being displayed.

Chart Filter Builder Settings

Enable/Disable

This attribute defines whether the user can disable fields or nestings in the filter builder, keeping them disabled until the application is reloaded.

Example of the disable button

Display filter condition

Defines whether the filter condition will be displayed while the user builds their filter.

When enabled, the condition is assembled as the user adds the filter’s conditions and logical operators. When disabled, this condition will not be displayed.

Modal Filter

Defines the location where the filter builder will open in the application.

When enabled, the filter will open in a modal. If disabled, the filter will be displayed in a div below the application’s toolbar. See below for examples of both behaviors.

Opening in a modal

Div below the toolbar

Save Search in Filter Build (Chart)

This feature allows the end-user to save his searches in a profile. You can create some rules, like to save the searches by user login.

 Save Filter Interface.

Save Filter Interface.

Related Links 🔗

- Search with buttons in line
- Advanced search with a predefined initial value
- Advanced search with relative periods
- Search with blocks
- Highlight option
- Macro sc_where_filter

Related Videos ▷

Select Fields for Filter Build in Chart

On this screen, the fields that will be available for use in the dynamic filter must be defined.

In the **left column**, all application fields and the **Search** field are displayed, which helps in locating the fields. In the **right column** are the fields selected to be displayed in the filter.

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

- - Move all fields to the right.
- - Moves only selected fields to the right
- - Moves only the selected fields to the left.
- - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the button.

> To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning

Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

> Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

Select Fields for Dynamic Search in Chart

On this screen, the fields that will be available for use in the dynamic filter must be defined.

In the **left column**, all application fields and the **Search** field are displayed, which helps in locating the fields. In the **right column** are the fields selected to be displayed in the filter.

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

- - Move all fields to the right.
- - Moves only selected fields to the right
- - Moves only the selected fields to the left.
- - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the button.

> To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning

Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

> Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

Chart Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid		
Procedure		
Chart		
Form		
Control		
Calendar		
Search		
PDF Report		

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven't manage fields.

Text

General Settings

This type of field allows the developer to create quickly inputs to insert and update data, where the final user can inform its data to be allocated in its database.

Type of Data

Defines the type of the field in the application. In this case we should select **Text**.

Label

Defines the title that will be displayed in the field when the application is executed. The terminology used in the interface is fundamental to the system have a good usability, whe should use names and familiar terms to the final user of the application, instead of using terms from the system.

For example, this text field that has the name **customerid**, the client would have a much better understanding of the functionality of the field when we define the label as **Customer Name**.

Not only a fixed text, the **Label** attribute allows the use of langs to define the field's title, making it possible to use your application in a multi language project.

SQL Type

Informs the type of the field in the database.

Field Behavior

Use autocomplete:

Field automatically turns into autocomplete according to existing values in the database.

Use Select2

Uses the new component for data selection, allowing searches within the select comboBox.

Width for the Select2

Sets a width for the area in the Select2.

Amount of characters

Sets the amount of characters to start the search.

Amount of rows

Sets the maximum number of rows to list the search result.

Width:

Defines a width in pixels for a result box.

Search options:

Defines the validation that will be made to fetch the search result.

Start equals to: Will return the records with the same start value as in the database.
Any part: Will return the records when exist the character in any part of the record.
End equals to: Will return the records with the same final value as in the database.

Position between values:

Defines the position that objects will be displayed.

Text Between Values

Text that will appear when using a filter condition between two values.

OnChange Submit:

Submit search on this field changing.

Show Condition:

To show or not the search condition. It only works if the search has at least one condition.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.

- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined, Advanced and Dynamic search
- Pivot Table with Filter
- Macro to_where_filter
- Highlight option

Related Videos

Integer

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Label:

It sets the title that will be displayed in the field when you run the application. The terminology used in the interface is fundamental for your system has a good usability. In this case, we recommend you to use names and terms familiar to the end user of the application, instead of using terms originated in the system.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Repeat value:

This option when enabled will allows you to repeat the field value if it is equal to the value of the previous record in the database.

Example:

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:

Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

```
SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}
```

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the

list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.

- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.

- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Ad
- Search with Advanced and Dynamic search
- Pivot Table with Filter
- Macro to_where_filter
- Highlight option

Related Videos ▷

Decimal

General Settings

Decimal field Configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Field Behavior

Use autocomplete:

Field automatically turns into autocomplete according to existing values in the database.

Use Select2

Uses the new component for data selection, allowing searches within the select comboBox.

Width for the Select2

Sets a width for the area in the Select2.

Amount of characters

Sets the amount of characters to start the search.

Amount of rows

Sets the maximum number of rows to list the search result.

Width:

Defines a width in pixels for a result box.

Search options:

Defines the validation that will be made to fetch the search result.

Start equals to: Will return the records with the same start value as in the database.
Any part: Will return the records when exist the character in any part of the record.
End equals to: Will return the records with the same final value as in the database.

Position between values:

Defines the position that objects will be displayed.

Text Between Values

Text that will appear when using a filter condition between two values.

OnChange Submit:

Submit search on this field changing.

Show Condition:

To show or not the search condition. It only works if the search has at least one condition.

Values format

Decimal Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator, Negative sign and Negative number format.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Advanced search with relative periods
- Refined Search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Currency

General Settings

Currency field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Currency, you can currency values to the field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Currency Field Behavior Interface of the Search Configuration.

- **Use auto-complete** : The field behaves as an auto-complete according to the values existing in the database.
- **Amount of characters** : Sets the amount of characters to start the search.
- **Amount of rows** : Sets the maximum number of rows to list the search result.
- **Width** : Sets the width in pixels for the result box.
- **Search options** : Defines the validation that will be made to fetch the search result.
- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Values format

Currency Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).

    - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the title of the field.
    - **Height** : To define a height for the title of the field.
- **CSS of the Field**
    - **Font** : Allows to choose the font type, that will be applied to the application field.
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the field.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
    - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the field.
    - **Height** : To define a height for the field.
- **CSS of the Input Object**
    - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the Input Object.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
    - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the Input Object.
    - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
    - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

    - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

    - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search

Related Videos

- Advanced search with highlighting
- Define Refined Search
- Search with Filter
- Pivot Table with Filter
- Advanced search with date periods
- Refined Advanced and Dynamic search
- Search with Slicers
- Pivot Table with Filter

Date

General Settings

- **Data Type** : Define the type of field for the application. When set to Date, you can inform a date.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Watermark**: Displays a watermark in the field input.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the datatype of field in the database.

Values format

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator attribute.
- **Date separator** : Allows you to inform the separator symbol for the date.
- **Display** : Select the format of the day for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **Date**. You need to use the characters **A**, **M** and **D** that correspond to **Year**, **Month** and **Day**.
- **Use Combo-box** : Allows you to select the date using a combo-box.
  - **Year as Combo** : Allows to use the year combo to select the date.
    - **Initial Year** : First year displayed in the combo.
    - **Actual Year +** : Display the current plus the amount of years informed.
- **Month in full textual** : Displays the Month format in Full.
- **Display Calendar** : Enables the a calendar icon beside the field, this allows to select the date from a calendar with the format already setup.
  - **New Calendar** : Defines if the JQuery calendar (New Calendar) is going to be displayed or the old format.
  - **Years Limit** : Amount of years displayed in the calendar.
  - **View week number** : Displays the number of the week in the application.
  - **Additional months** : Displays the additional months of the calendar.
  - **Show Combo year and month** : Displays the year and month of the calendar in the combo box.

Field Behavior

- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Special Conditions

- **All Period** : Searches for all periods of dates.
- **Today** : Searches in today’s date.
- **Yesterday** : Searches in yesterday’s date.
- **Last 7 days** : Searches the last 7 days. Ex: ((01/01/2017 01/07/2017).
- **This month** : Searches the dates from the first day of the current month.
- **Last month** : Searches the dates from the first day of lasts month.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of

Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refine Search
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Time

General Settings

Time field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Time, you can inform a time to this field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Values format

Time Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator and time separator attributes.
- **Time separator** : Allows you to inform the separator symbol for the time.
- **Display** : Select the format of the time for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **TIME**. You need to use the characters **HH**, **II**, and **SS** that correspond to **Day**, **Hour**, **Minutes** and **Seconds**.

Field Behavior

Time Field Behavior Interface of the Search Configuration.

- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.

- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Search with Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos

Datetime

General Settings

Datetime field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Datetime, you can inform a date and time to this field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Values format

Datetime Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator and time separator attributes.
- **Date separator** : Allows you to inform the separator symbol for the date.
- **Time separator** : Allows you to inform the separator symbol for the time.
- **Display** : Select the format of the day/time for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **DATETIME**. You need to use the characters **A**, **M**, **D** , **HH**, **II**, and **SS** that correspond to **Year**, **Month**, **Day**, **Hour**, **Minutes** and **Seconds**.
- **Use Combo-box** : Allows you to select the date using a combo-box.
  - **Year as Combo** : Allows to use the year combo to select the date.
    - **Initial Year** : First year displayed in the combo.
    - **Actual Year +** : Display the current plus the amount of years informed.
- **Month in full textual** : Displays the Month format in Full.
- **Display Calendar** : Enables the a calendar icon beside the field, this allows to select the date from a calendar with the format already setup.
  - **New Calendar** : Defines if the JQuery calendar (New Calendar) is going to be displayed or the old format.
  - **Years Limit** : Amount of years displayed in the calendar.
  - **View week number** : Displays the number of the week in the application.
  - **Additional months** : Displays the additional months of the calendar.
  - **Show Combo year and month** : Displays the year and month of the calendar in the combo box.

Field Behavior

Date Field Behavior Interface of the Search Configuration.

- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Special Conditions

Datetime Field Special Conditions.

- **All Period** : Searches for all periods of dates.
- **Today** : Searches in todays date.
- **Yesterday** : Searches in yesterdays date.
- **Last 7 days** : Searches the last 7 days. Ex: ((01/01/2017 01/07/2017).
- **This month** : Searches the dates from the first day of the current month.
- **Last month** : Searches the dates from the first day of lasts month.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**

- **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links 🔗

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos ▷

Select

General Settings

Select field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Select, you can select multiple option from a combo box (Select Field).
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Select Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

  Selecting the lookup type.

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimitation.

      - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Height** : Defines the height for the select object.
    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to

a value informed manually (Title internal value).

- **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
- **Title** : Text displayed on the title of the Select object.

- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.
- **Link** : Allows to create a link to another form allowing to manipulate the list displayed on the select field. After the manipulation, the select object it updated automatically.

- **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Height** : Defines the height for the select object.
  - **Multiple Values (delimiter)**

    You can store various values for the select field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Height** : Defines the height for the select object.
  - **Multiple Values (position)**

    Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of

bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Height** : Defines the height for the select object.

- **Multiple Values (binary)**

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

Editing Lookup Configuration Multiple Values (Binary)

- **Label** : Text that will be displayed in the item list of the select field.
- **Height** : Defines the height for the select object.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  Displaying the original lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Lookup Method - Actual value

This lookup is used to list all the values in the selected field.

This lookup will apply a “distinct” to your SQL query.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Dynamic Search
- Advanced search with relative periods
- Refined Search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Double Select

General Settings

Double Select field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Double Select, your allowed to have multiple options selected.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Double Select Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, getting these values from the database.

Lookup Settings Display for the field. *Automatic Lookup Interface..*

- **SQL Select Statement** : Defines the SQL command that will get the values displayed on the form field. The command needs to have the following structure:

  ```
  SELECT key_field , display_field FROM table
  ```

  The value of the key_field will be stored in the table field.

- **Height** : Set the height(lines) of the field interface.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter than (;).
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**

- **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links 🔗

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos ▷

Check box

General Settings

Check box field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Check box, your allowed to have multiple options selected.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Check box Field Behavior Interface of the Search Configuration.

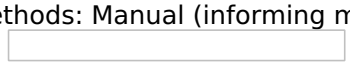
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).  *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the CheckBox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
    - **Option check all** : Displays two options on the field to check and uncheck all.

    - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Columns** : Set the amount of columns, for the list of items.
  - **Multiple Values (delimiter)**

    You can store various values for the checkBox field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Columns** : Set amount of columns, for the list of items.
  - **Multiple Values (position)**

    Stores a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

    For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

    The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

    Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the

following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Columns** : Set the amount of columns, for the list of items.

- **Multiple Values (binary)**

  Stores a diversity of information selected on the field in only one value.

  In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  To store the data, it is done an internal sum of the decimal number. See the examples below.

  Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 2 (Culture) = **3**

  Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

  Setting up Multiple Values (Binary)

  - **Label** : Text that will be displayed in the item list of the checkBox.
  - **Columns** : Allows you to inform the amount of columns, for the list of items.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Saves all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Refreshes the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Option check all** : Displays two options on the field to check and uncheck all.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio,

Select, Text, etc
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

    Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

    Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

    Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos

Radio

General Settings

Radio field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Radio, your allowed to select one of the options listed.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Radio Field Behavior Interface of the Search Configuration.

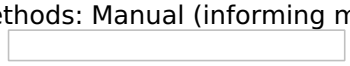
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).  *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Columns** : Allows you to inform the amount of columns, for the list of items.

    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

      - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
      - **Title** : Text displayed on the title of the Select object.

- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

  This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

  Manual Lookup Interface.

  - **Label** : Text that will be displayed in the item list of the radio.
  - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
  - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
  - **Insert Button** : Adds to the list the values informed on the fields Label and Value.
  - **Update Button** : Modifies the attributes of the selected item.
  - **Remove Button** : Remove the selected item from the list.
  - **Clear Button** : Clear the Fields.
  - **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on load lookup definition.
  - **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
  - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

    - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
    - **Title** : Text displayed on the title of the Select object.
  - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
  - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

    Displaying the original lookup.
  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Defined Search
- Quicksearch and Highlight option
- Search Refined by Advanced and Dynamic search
- Pivot Table with Filter
- Macro se_where_filter
- Highlight option

Related Videos

Text Auto-Complete

General Settings

Text Auto-Complete field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Text auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal Text for the data.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Text Auto-Complete Field Behavior Interface of the Search Configuration.

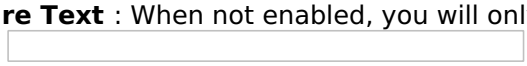
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **Case Settings** : Allows to convert the letters of the field when losing the focus. The options are:
  - **Upper Case** : All in Upper Case.
  - **Lower Case** : All in Lower Case.
  - **Capitalize first word** : Capitalizes the first letter of the word.
  - **Capitalize all words** : Capitalizes the first letter of all the words.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below.  *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Defined Search
- Quicksearch and Highlight option
- Search Refined by Advanced and Dynamic search
- Pivot Table with Filter
- Macro se_where_filter
- Highlight option

Related Videos

Number Auto-Complete

General Settings

![Number Auto-Complete Field Behavior Interface of the Search Configuration.][filtro_cons_número_auto]
Number Auto-Complete Field Behavior Interface of the Search Configuration.

- **Data Type** : Define the type of field for the application. When set to Number auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal number for the data.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Number Auto-Complete Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Values format

Number Auto-Complete Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator, Negative sign and Negative number format.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below. *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps

the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value

Related Videos

- Dynamic Search
- Advanced search with relative periods
- Quick search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Date Comparison

Special field that allows you to change the functioning of a date type field in the summary filter, enabling the comparison of metrics over time. To use this type of field, it is necessary to have a date type field created, and later change its type to Date Comparison.

> It is only available in the advanced filter and its operation is exclusive to the query summary module.

Example of the field in the summary filter

Example of the Generated summary

When using the comparison, the summary will be assembled showing the values for the periods informed.

General Configuration

In this block are the general settings of the field, and where it will be 'transformed' into a comparative data field, simply access the settings of a data field and change the *Data* type to **Date comparison**

Data type of the field.

This attribute defines the type of field in the application. This definition implies the configuration options that will be displayed to the developer as well as the behavior of the field when running the application.

During the application creation process, Scriptcase maps the fields of the table(s) used and defines the data type of the fields according to their SQL type.

For example
Fields of type varchar in the table will be defined as text in the **Data Type** attribute.

> It is possible to change the field's data type, however it is important to take into account the **SQLType**, displayed in the General Configuration block, to define the field's data type.

Search Label

Defines the title of the column displayed in the executed application and can be defined using: Lang, Text or Global variable.

Lang

The use of lang in the attribute definition is recommended for creating multilingual systems. The langs must be previously defined using the option Traduzir aplicações or through Dicionário de Dados.

Fixed text

In this case, the column title will not be changed due to user interaction, such as changing the application language, for example.

It is a good option for systems created with just one language.

Global variable

The use of global variables allows you to change the column title according to your system business rule.

> We also provide the macro sc_label which allows you to change the column title regardless of how the label attribute was defined.

Example of a column title

Use same label used on the Grid

This option, when activated, allows you to use the title of the field defined in the grid, in this case the value defined in the **Label filter** attribute will be disregarded.

Value Formatting

Regional Settings

Allows you to apply regional date formatting settings to the field, according to the application language. When disabling the use of *regional settings*, some settings will be displayed such as: **Date separator**, **First Day** and **Display**.

Check below for more details about each configuration option.

> It is possible to change regional settings in general, allowing the use of the option and the same configuration in all your projects, according to the language used. For this configuration, access the Menu *Locales* > *Configurações Regionais*

Date separator

Defines the character that will be used to visually separate the parts of the date (day, month and year). The characters normally used are: slashes (“/”), hyphens (“-“) and periods (.).

Alguns exemplos

- **slashes (/)**: 25/10/2023
- **hyphens (-)**: 25-10-2023
- **periods (.)**: 25.10.2023

> This option will be displayed if the **Use regional settings** attribute is disabled.

First day

Defines the first day of the week to be displayed in the calendar in the date field, for selecting dates.

First day change example

In this example, the *First day* was defined as Monday, changing the order in which the days of the week are displayed in the field calendar.

> This option will be displayed if the **Use regional settings** attribute is disabled.

Display

Defines the date display format by selecting one of the listed formats.

The options are:

- dd-mm-aaaa
- mm-dd-aaaa
- aaaa-mm-dd

> This option will be displayed if the **Use regional settings** attribute is disabled.

View week number

Defines whether the number of weeks will be displayed in the date field calendar.

Example of a calendar with number of weeks

Special Conditions

Defines the special conditions that will be displayed in the date field calendar, see the example below.

Example of special conditions in the date field calendar

If no interval is enabled, the calendar in the date field will not have any special conditions.

Example of the date field without special conditions

Select Values

In this attribute, the special conditions that will be listed as an option in the application must be defined. The predefined intervals are grouped according to the period they refer to. For example, the condition *Last week* is in the tab **Week**.

In each of the tabs it is also possible to create personalized intervals using the button Add new range.

In the tab , all selected ranges will be listed, so that the display order can be changed according to the user's needs.

Sorting tab

In this tab, all selected conditions will be listed, allowing the intervals to be reordered according to the system's needs. You can also remove conditions one by one or using the button **Remove all**.

Add New Range

Clicking on *add new range* will open a screen with some options for adding new ranges. To add an interval, we must select the type (next or last) and enter the period (value).

For example, adding a range in the year tab, defined *Last* in the attribute as **Type** and reporting *2* in the attribute **Value** a range will be created: **Last two years**

> This button is available on the **Year**, **Quarter**, **Month**, **Week** and **Day** tabs.

Type

Defines whether the created interval will refer to a future period (next) or a previous period (last).

- **Next** - Defines that the created interval will refer to a future date, for example, **Next year**.
- **Last** - Defines that the range will refer to a previous date, for example, **Last Year**.

Value

Sets the amount of time for the interval.

> This field only accepts positive integer values greater than zero.

Title

Defines the text that will be displayed in the application. When creating a new range, a lang is automatically

generated, it is also possible to use a fixed text.

Include Current

This option changes how the created interval works, adding the current period to the interval.

For example, adding an interval to the year tab, defining *Next* in the **Type** attribute and entering *2* in the **Valor** attribute and checking the **Include current** option will create an interval: **Next 2 years from current**.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse

leaves the help icon.

- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

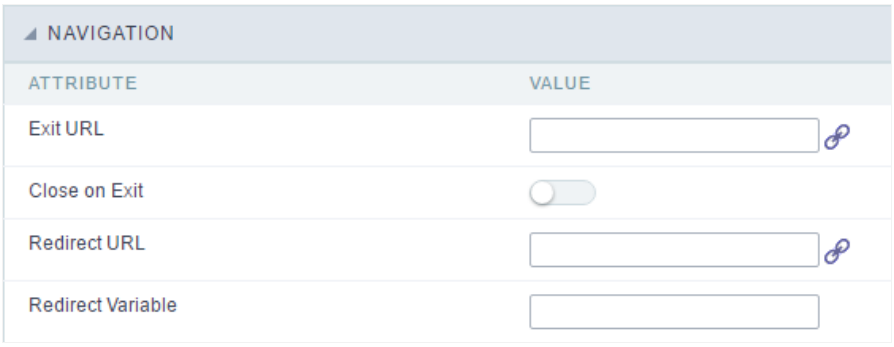
Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Application Settings (Chart)

Settings

Navigation

This interface allows defining the navigating behavior of the application



Navigation Interface.

Exit URL

URL to where the user goes when he clicks on the “exit” button.

Close on Exit

Close the browser window when the user clicks on the “exit” button.

Redirect URL

Redirect to another URL in case there aren’t any global variables available.

Redirect Variable

Creates a variable with the application name and sends it to the redirected application.

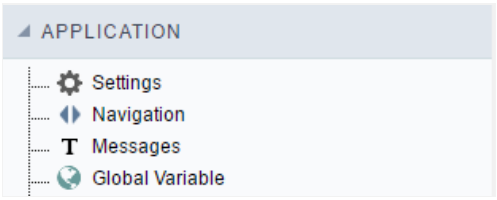
Global Variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**“**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target=”_blank”} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



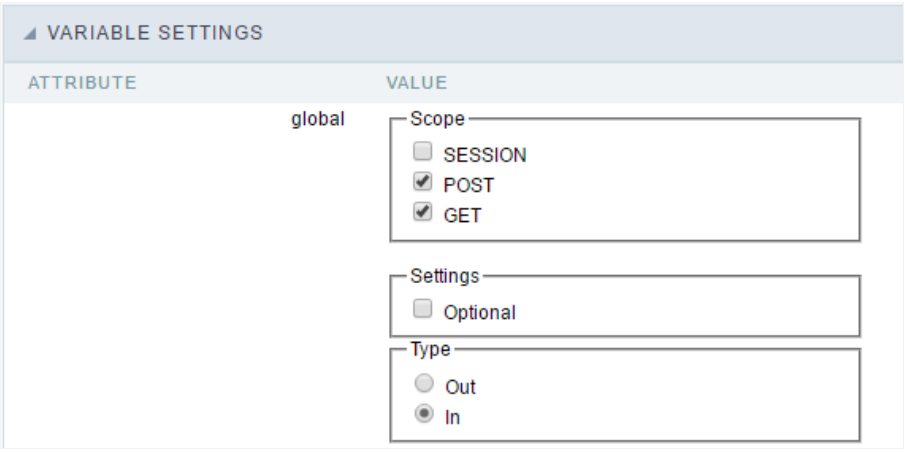
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

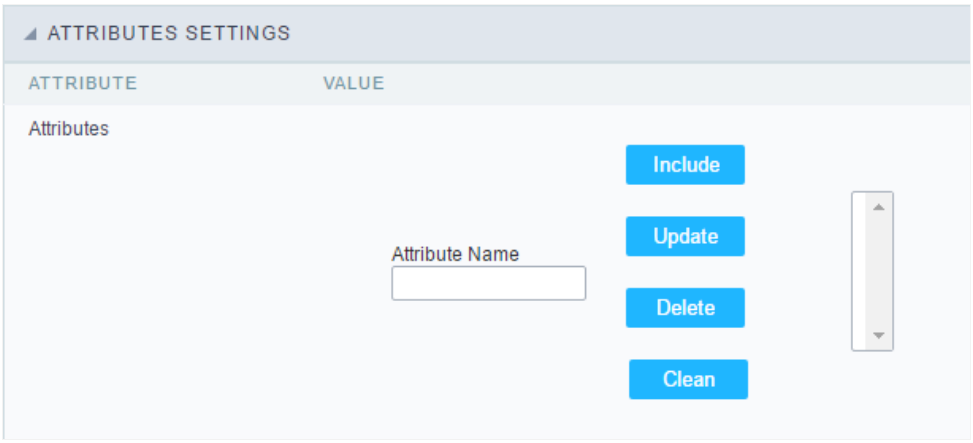
Indicates where the application is using the variable.

Chart Programming

In this version of ScriptCase is incorporated the concept of programming, with the use of attributes, methods, resources and libraries. In the previous version it was already possible to create business rules in applications using this concept the big difference now is that this can be done in a more organized, facilitating both the development as the understanding of the rule by another developer.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal Libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

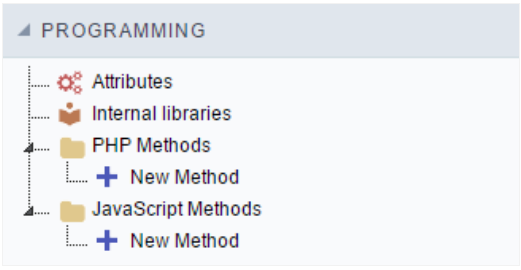
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

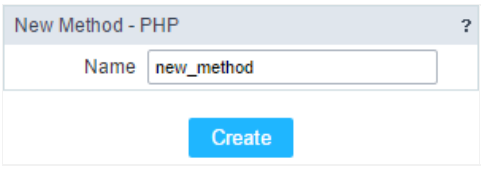
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

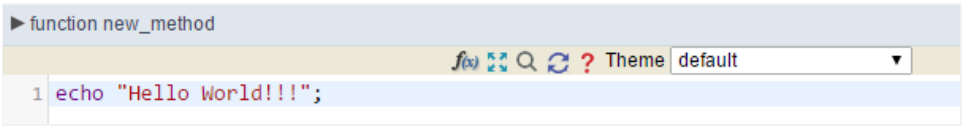


- Give the method a name and click Create. As in the image below.

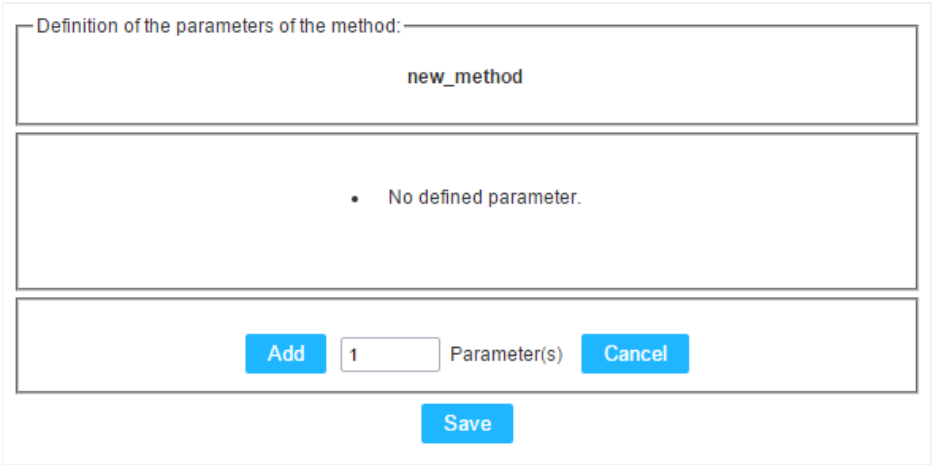
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



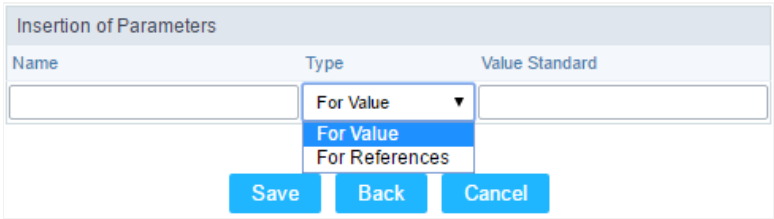
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

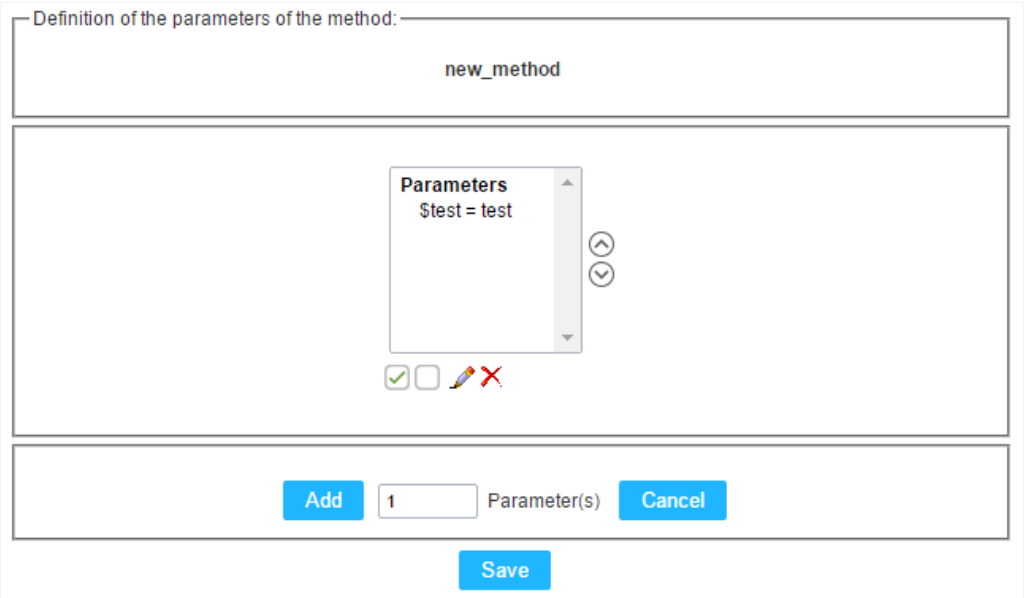


- **Name** : Type in the variable's name.

- **Type** : Selecting the type of variables: For Value or For Reference.
- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:



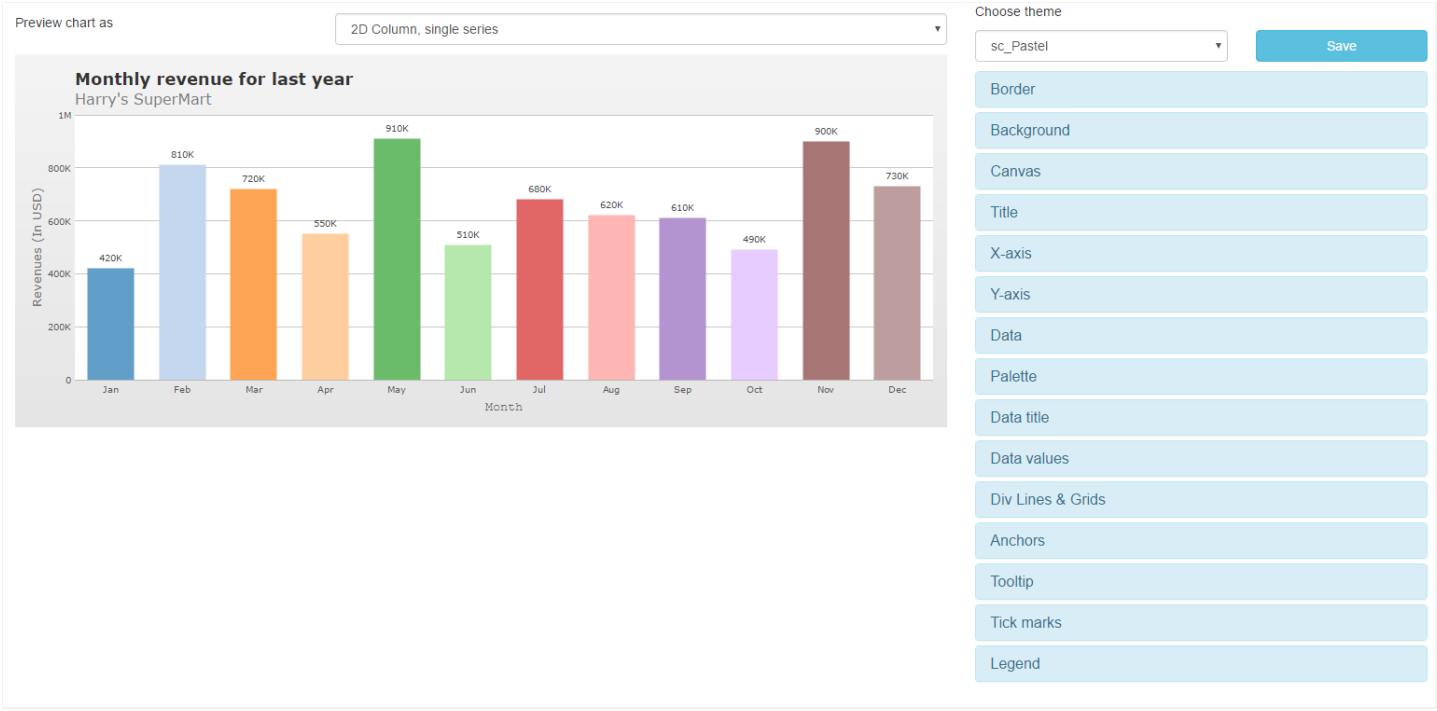
- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

Chart Theme

Preview

In this area you can customize the chart themes, you will be able to preview how it will look like according to each chart type.

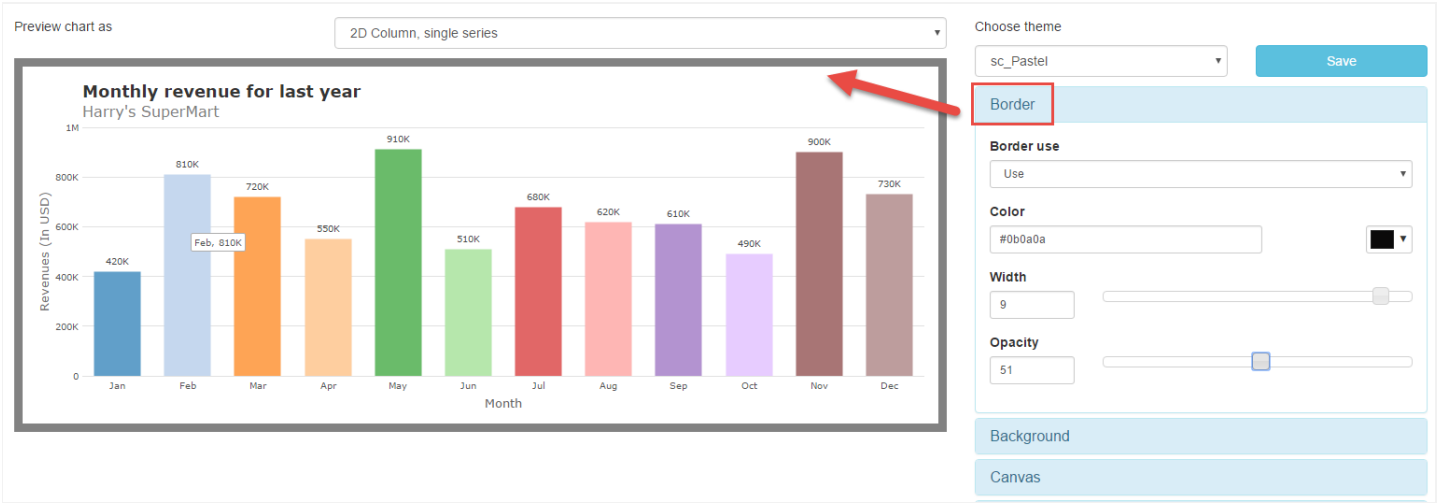
If you customize the theme using this area, it will available only for the current application only. If you need to customize the theme and apply to other application, project and users, then you will need to use the “Chart themes” option within the main menu “Layout > Chart themes”. There you will find the same interface with the option to save the changes as “user level, project level or Scriptcase level”



Themes preview

Border

Activate this function to add a border around the chart area.



Border.

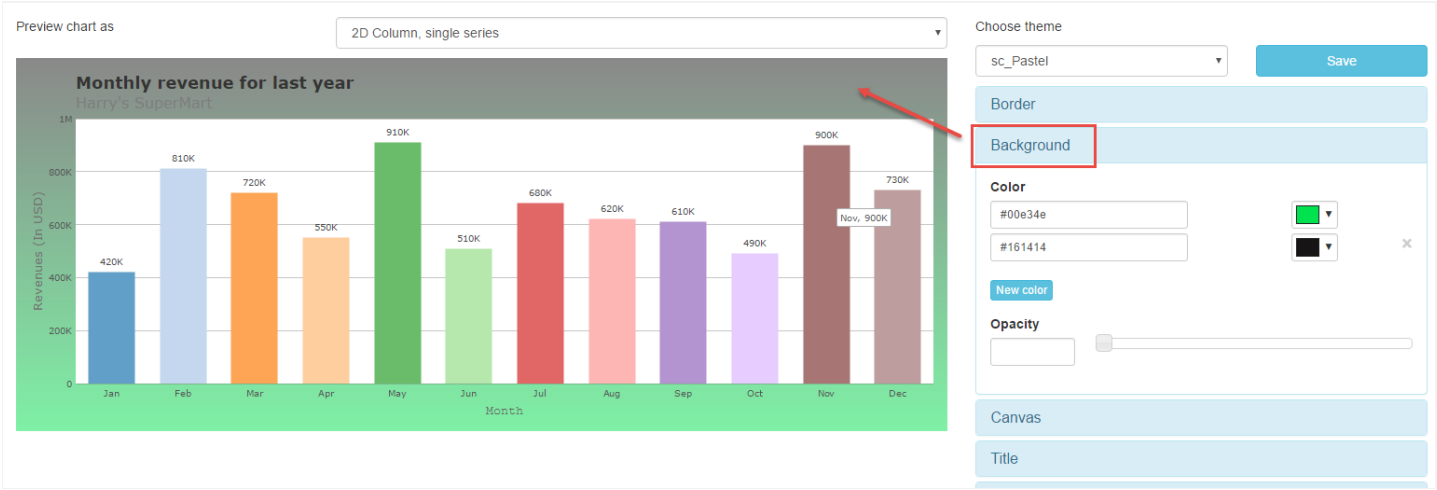
- **Border use** : Not informed, use or do not use.

- **Not informed**: This option will use the default option of the theme.
- **Use** : Select the border as active, you will need to set the width.
- **Do not use** : to remove or do not use the border.

- **Color** : Select the border color, and may report a hexadecimal value or select using the color picker.
- **Width** : Sets the border width in pixels.
- **Opacity**: Sets the border transparency.

Background

Options to configure the background color of the chart.

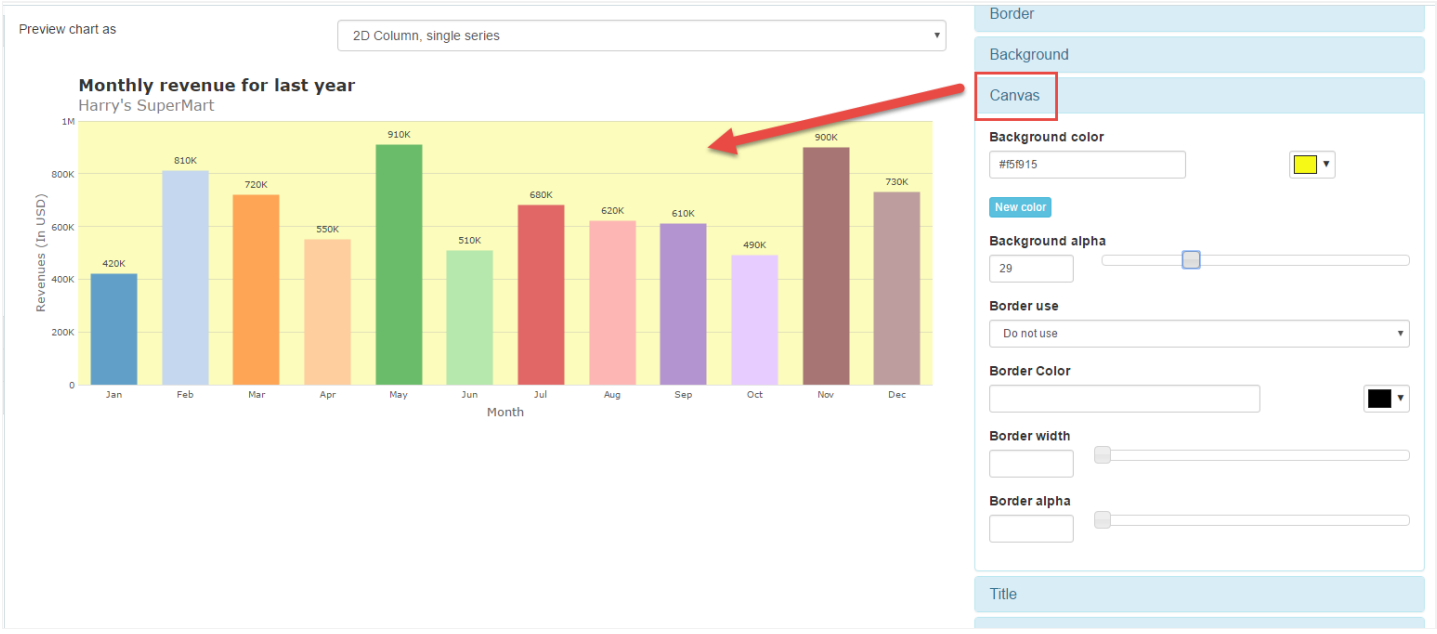


Background

- **Color** : Select the background color, and may report a hexadecimal value or select using the color picker.
- **New color** : Adds a new color for the background.
- **Opacity** : Select background transparency.

Canvas

Options to configure the chart canvas.



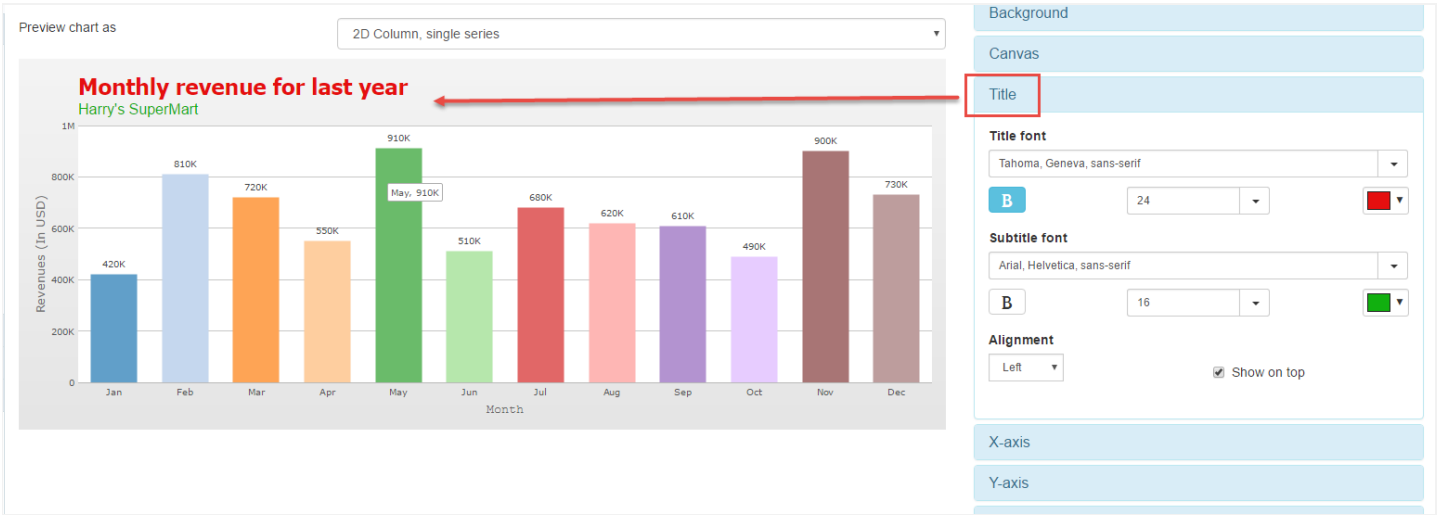
Canvas

- **Background color** : Selects the color of the canvas, and may report a value hexadecimal or choose picker.

- **New color** : Adds a new color option.
- **Background alpha** : Selects the background transparency.
- **Border use** : Not informed, use or do not use.
  - **Not informed**: This option will use the default option of the theme.
  - **Use** : Selects the border as active, you will need to set the width.
  - **Do not use** : To remove or to do not use the border.
- **Color** : Selects the canvas color, and may report a hexadecimal value or select using the color picker.
- **Width** : Sets the border width in pixels.
- **Opacity**: Sets the canvas transparency.

Title

Options to configure the formatting of the title.

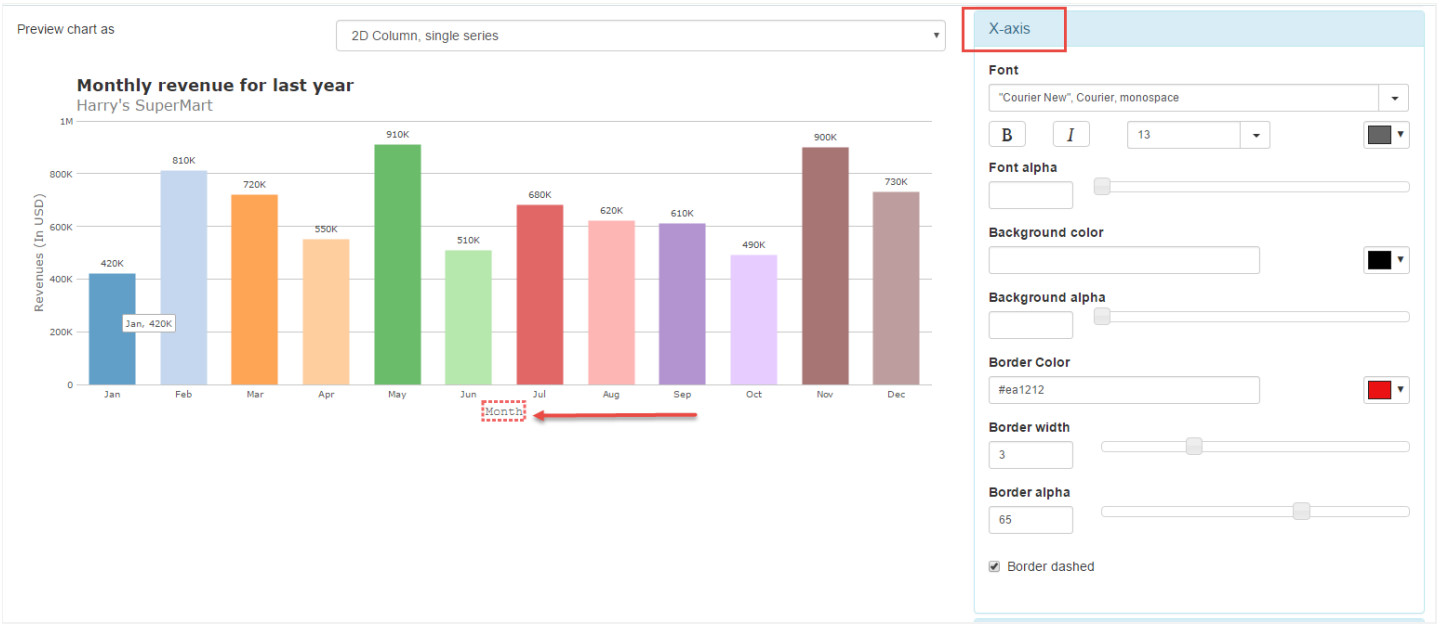


Title

- **Title font** : Selects the font, size, color, and formatting of the title.
- **Subtitle font** : Selects the font, size, color and format of the subtitle to left, center or right.
- **Alignment** : Selects the position of the title and subtitle.
- **Show on top** : If this option is selected the title and the subtitle will be positioned above the chart, otherwise will be positioned below.

Axes

This option configures the formatting of the texts of the x and y axis

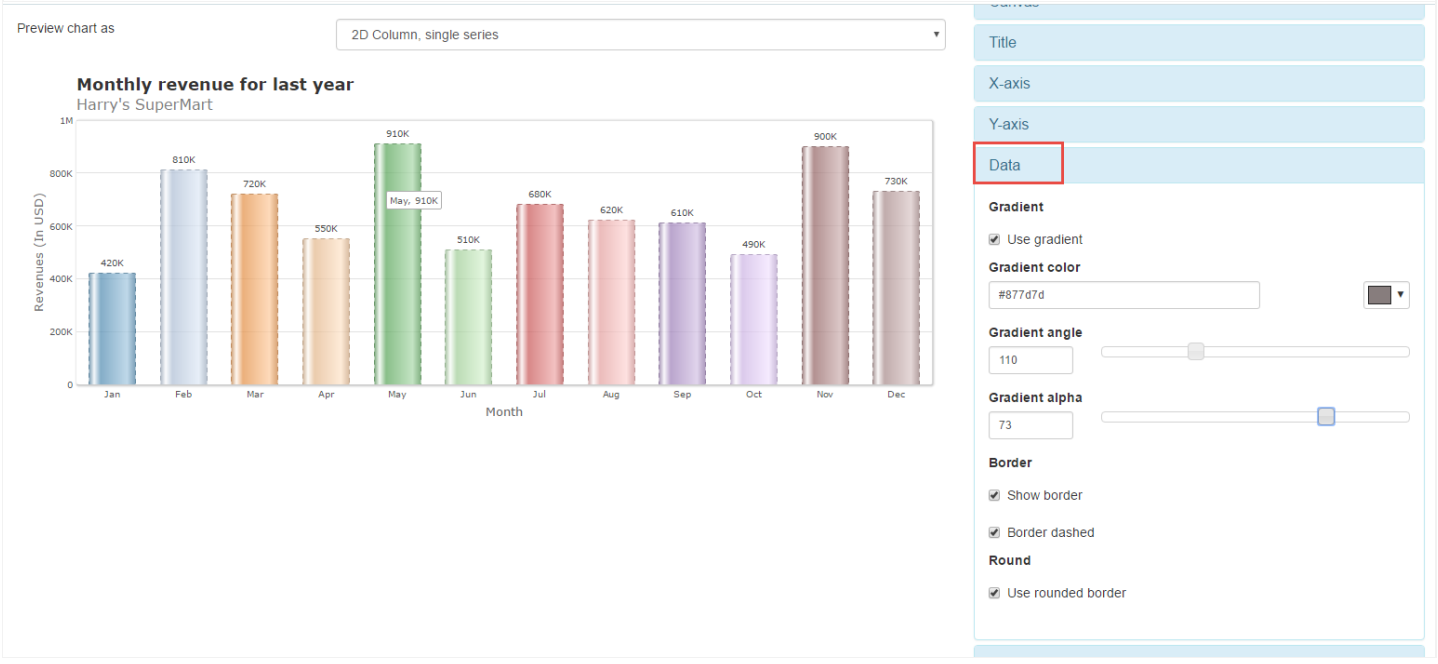


Axes

- **Font** : Selects the font, size, color, and formatting of the axis.
- **Background color** : Selects the background color, and may report a hexadecimal value or select using the color picker.
- **Background alpha** : Selects the transparency of the background color of the axis.
- **Border Color** : Selects the color, and may report a hexadecimal value or select using the color picker.
- **Border width** : Sets the width in pixels.
- **Border alpha**: Sets the transparency.
- **Border dashed**: If this option is selected, the border is dashed, this option is only valid if the option “Display” is checked.

Data

Options to set the chart’s data (bank records).



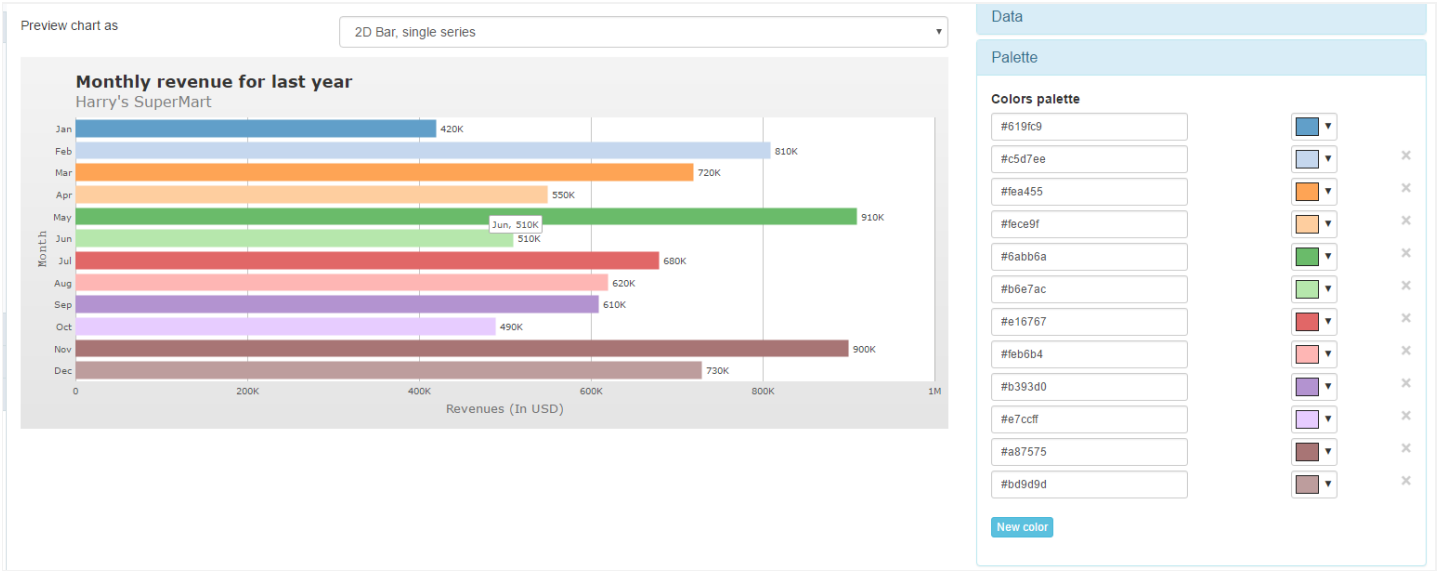
Data

- **Use gradient** : Enables the data gradient.
- **Gradient color** : Selects the gradient color, and may add a hexadecimal value or select using the color picker.
- **Gradient angle** : Selects the angle of the gradient.

- **Gradient alpha** : Selects the gradient transparency.
- **Show border** : If this option is selected is activated the border in the data.
- **Border dashed** : If this option is selected, the border is dashed, this option is only valid if the option “Display” is checked.
- **Use rounded border** : This option leaves the rounded border.

Palette

Options to choose the colors of each chart palette.

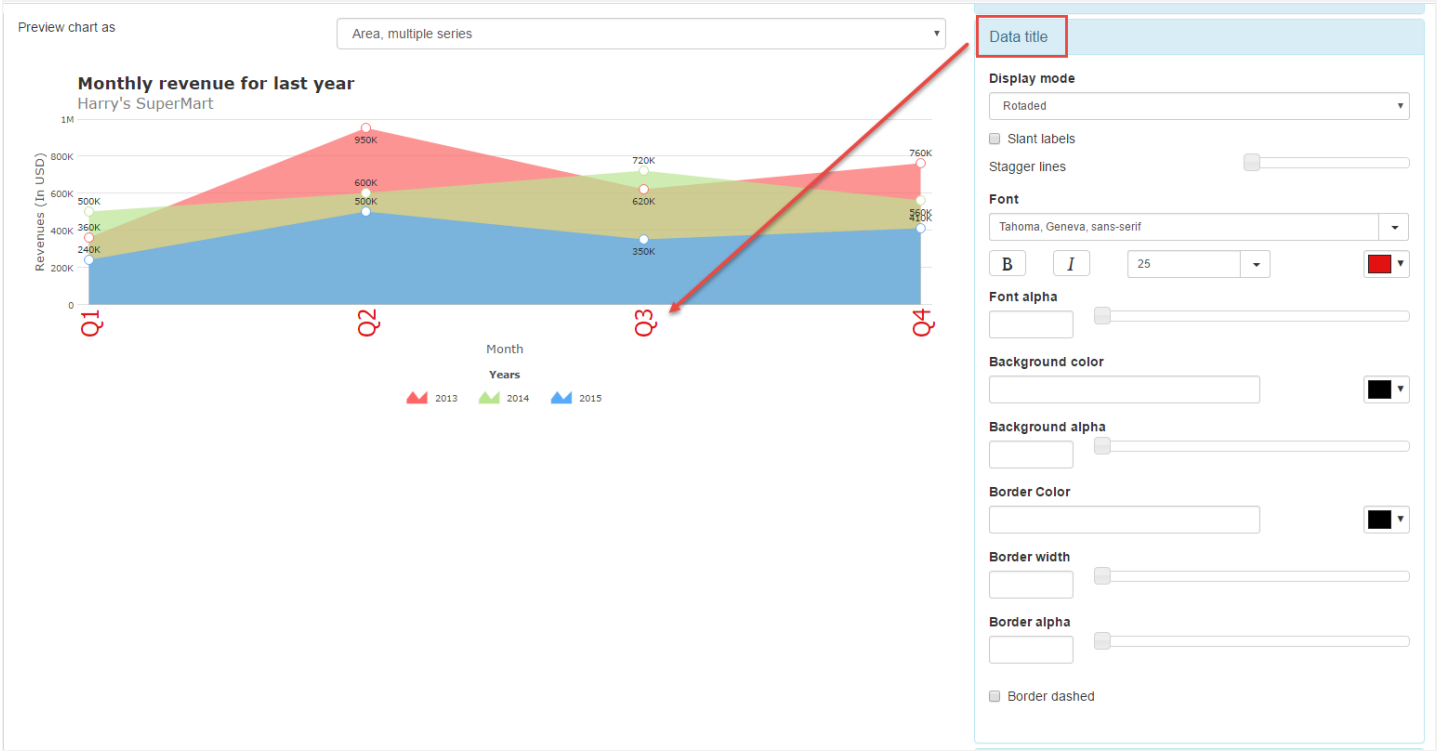


palette

- **Colors palette** : Select the color of each chart palette.

Data title

Options to configure title formatting of data.

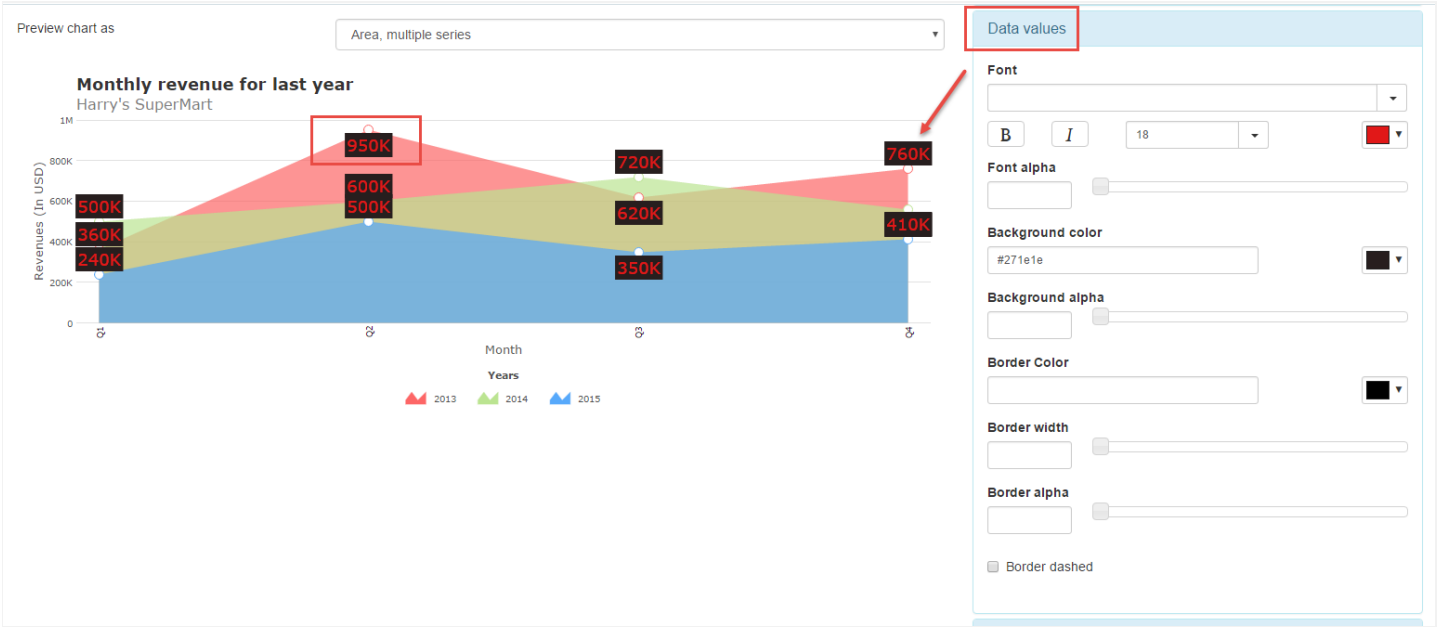


Data title

- **Display mode** : Selects the display mode to “automatic”, “rotated”, “line break”, “in levels” and “None”, if the Automatic option is selected, select the default option.
- **Slant labels** : Selects the number of levels to the data.
- **Font** : Selects the font, and your formatting such as bold, italic, size and color of data.
- **Font alpha** : Selects the transparency of data title font.
- **Background color** : Select the background color, and may add a hexadecimal value or select using the color picker.
- **Background alpha** : Selects the transparency of the background color of the title of data.
- **Border width** : Sets the width in pixels.
- **Border alpha**: Sets the transparency.
- **Border dashed**: If this option is selected, the border is dashed, this option is only valid if the option “Display” is checked.

Data values

Options to configure the formatting of data values.

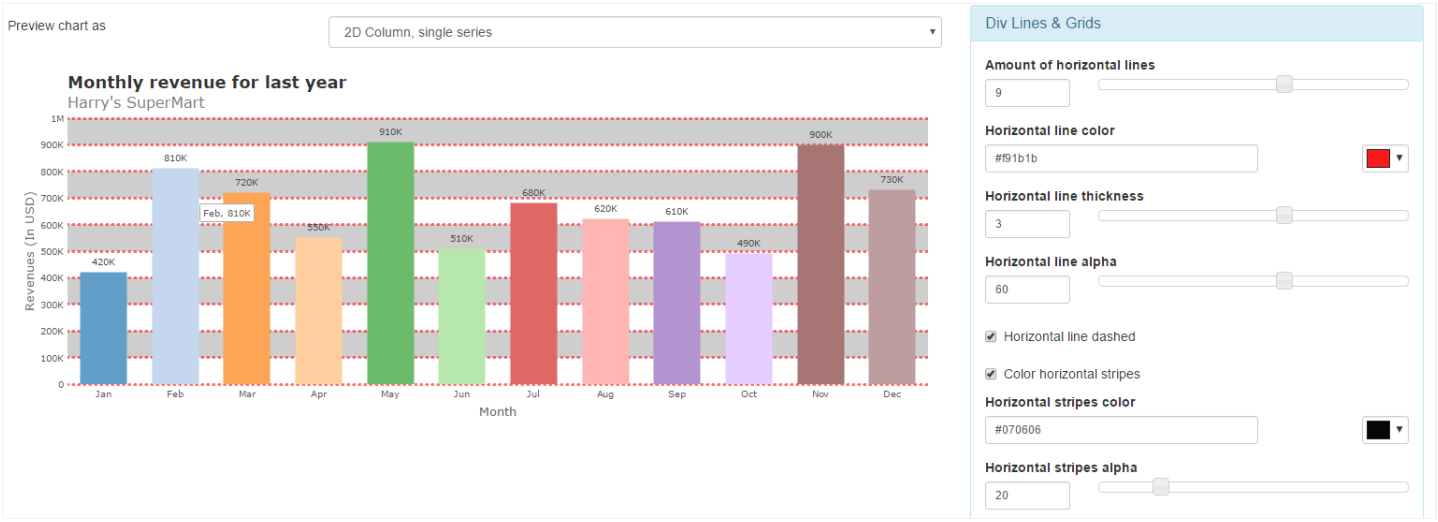


Data values

- **Font** : Selects the font, and your formatting such as bold, italic, size and color of the data values.
- **Font alpha** : Select the transparency of the source of the data values.
- **Background color** : Select the background color, and may add a hexadecimal value or select using the color picker.
- **Background alpha** : Selects the transparency of the background color of the title of data.
- **Border Color** : Selects the border color for the data values.
- **Border width** : Selects the border width for the data values.
- **Border alpha** : Selects the transparency of the color of the border of the data values.
- **Border dashed** : If this option is selected, the border is dashed, this option is only valid if the option “Display” is checked.

Div Lines and Grids

Options to configure the Division of horizontal and vertical lines on the graph.

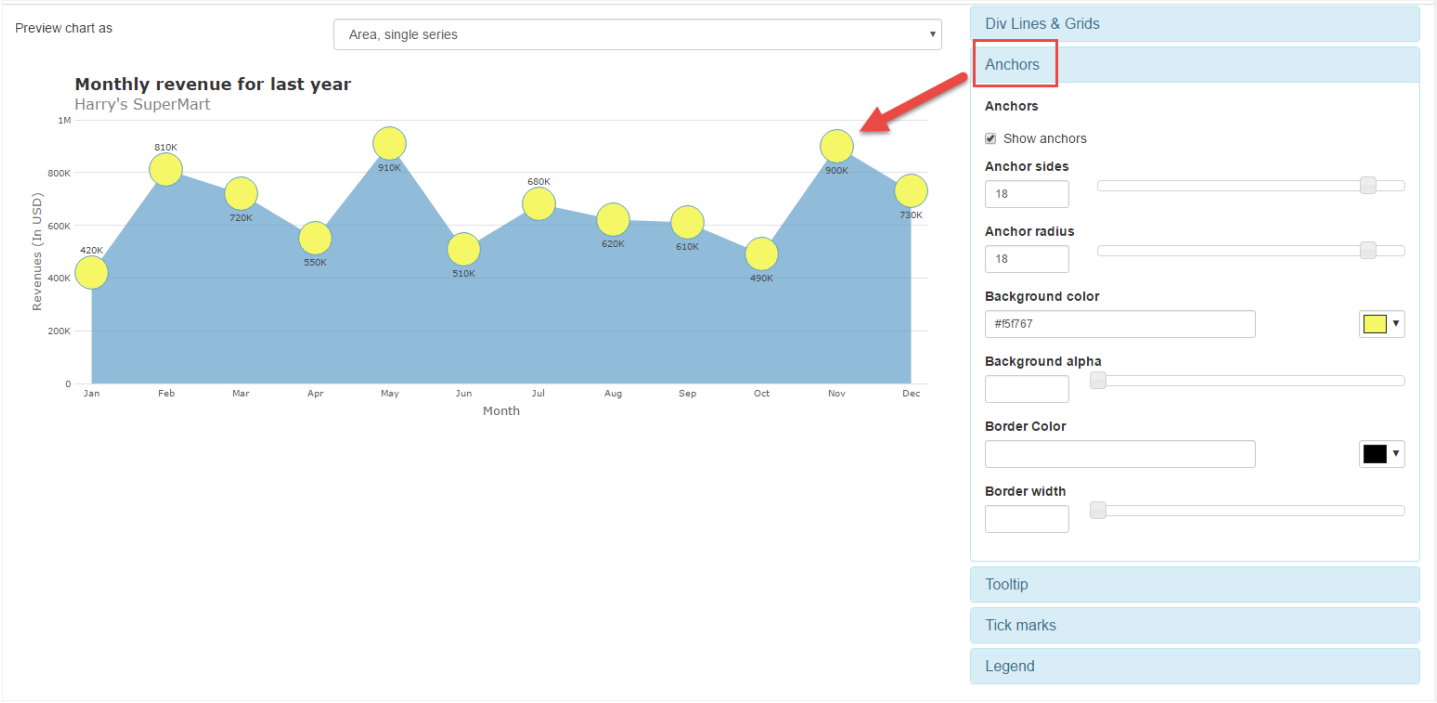


Div Lines and Grids

- **Amount of horizontal lines** : Selects the amount of horizontal lines that will be shown in the chart.
- **Horizontal line color** : Selects the horizontal line color.
- **Horizontal line thickness** : Selects the thicker horizontal line.
- **Horizontal line alpha** : Selects transparency horizontal line
- **Horizontal line dashed** : If this option is selected will horizontal line plot.
- **Color horizontal stripes** : If this option is selected you can choose the horizontal line color.
- **Horizontal stripes color** : Selects the color you wish to case “Coloring horizontal Ribbon” is checked.
- **Horizontal stripes alpha** : Selects transparency horizontal ribbon.
- **Amount of vertical lines** : Amount of vertical lines
- **Vertical line color** : Selects the color of the vertical line.
- **Vertical line thickness** : Selects the vertical line thicker.
- **Vertical line alpha** : Select the transparency of the vertical track.
- **Vertical line dashed** : If this option is selected will draw the vertical line.
- **Color vertical stripes** : If this option is selected the developer can choose the color of the vertical line.
- **Vertical stripes color** : Selects the color you wish to case “Color vertical stripes” is checked.
- **Vertical stripes alpha** : Select the transparency of the vertical stripes.

Anchors

Options to configure anchors on the chart.

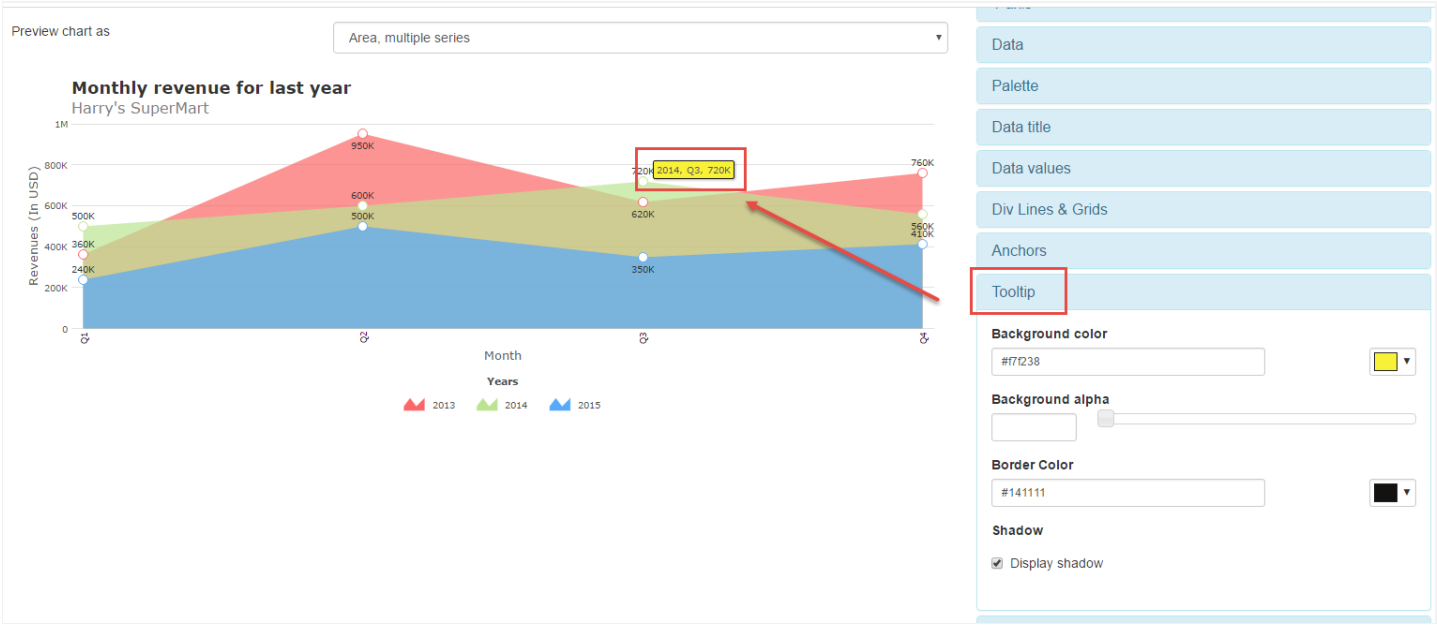


Anchors

- **Show anchors** : Enables the anchors in the chart.
- **Anchor sides** : Select the width of the anchor.
- **Anchor radius** : Selects the size of anchor beam.
- **Background color** : Select the color, and may add a hexadecimal value or select using the color picker.
- **Background alpha** : Selects the transparency of the anchor.
- **Border Color** : Select the anchors border color, and may add a hexadecimal value or select using the color picker.
- **Border width** : Select the border width of the anchor.

Tooltip

Options to configure the tooltips in the chart.



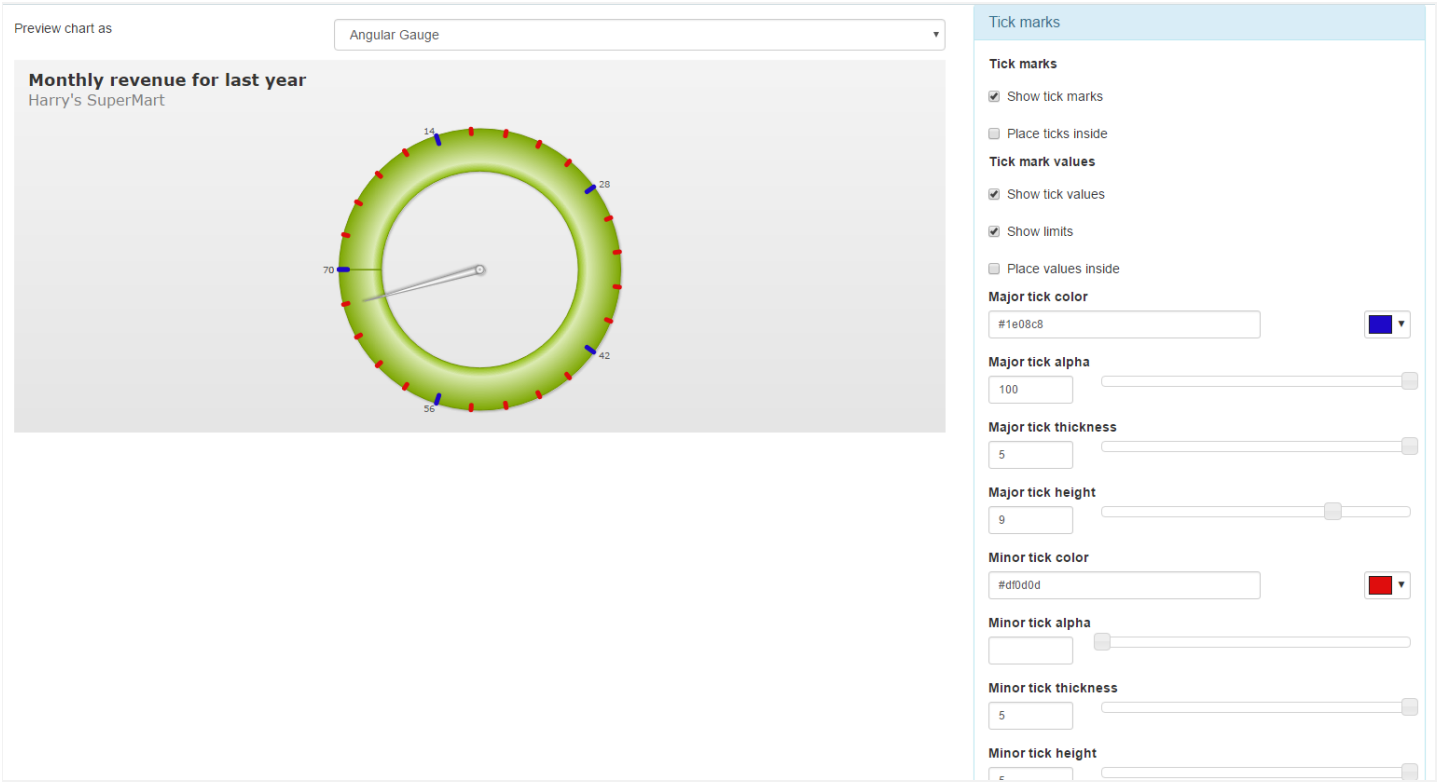
Tooltip

- **Background color** : Select the tooltip color, and may add a hexadecimal value or select using the color picker.
- **Background alpha** : Select the transparency of the background of the tooltip.

- **Border Color** : Select the tooltip border color, and may add a hexadecimal value or select using the color picker..
- **Shadow** : Enable the shadow in the tooltip.

Tick marks

Option to the configure chart marks, this option is only possible for the “Gauge” charts.

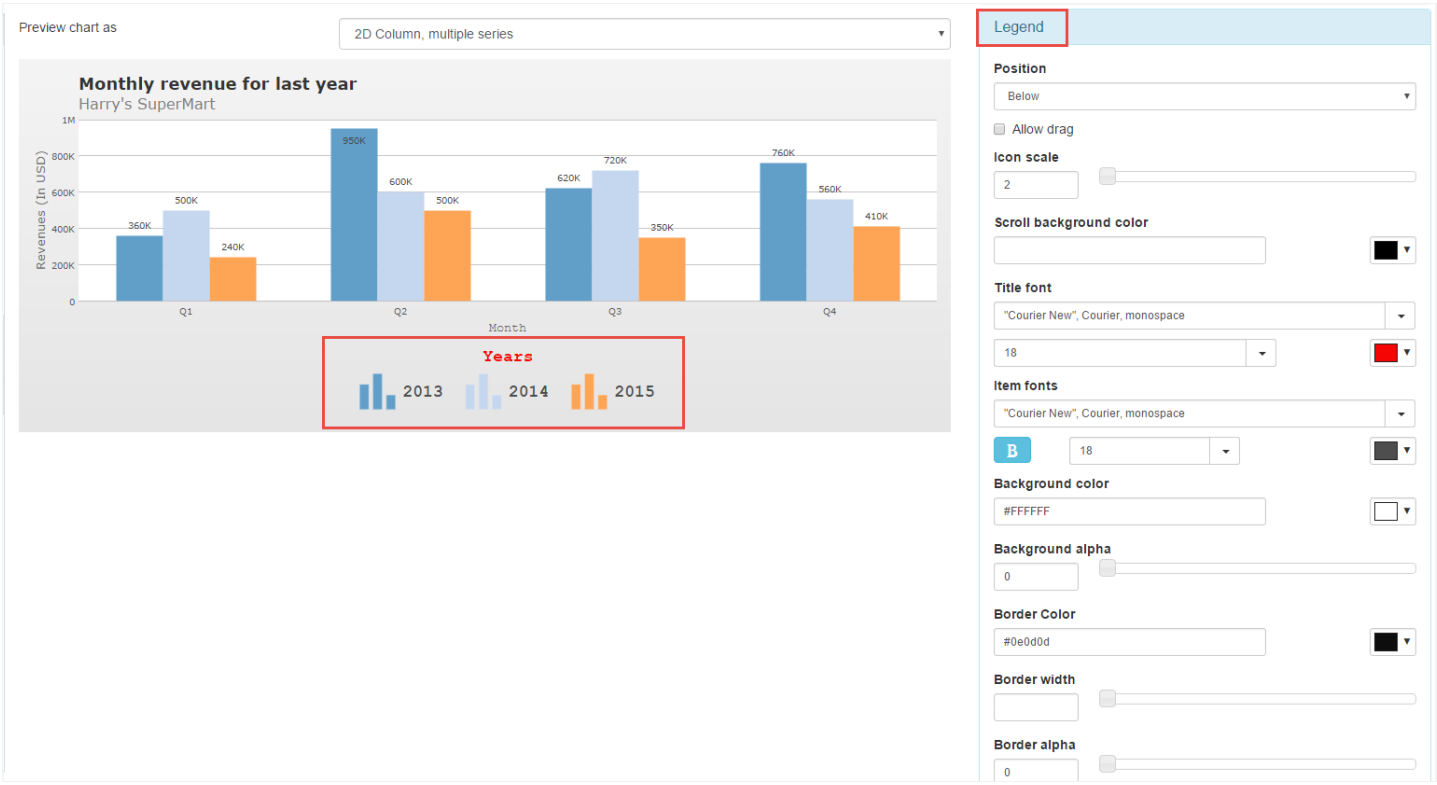


Tick marks

- **Show tick marks** : Enables the tick marks on the chart.
- **Place ticks inside** : If this option is checked the marks of the chart will tile chart poker.
- **Show tick values** : Show tick values
- **Show limits** : If this option is selected will show the chart limits.
- **Place values inside** : If this option is selected will show the values in the chart
- **Major tick color** : Major tick color.
- **Major tick alpha** : Major tick transparency
- **Major tick thickness** : Major tick thickness.
- **Major tick height** : Select a height for the major tick.
- **Minor tick color** : Select a color for the minor tick
- **Minor tick alpha** : Select the transparency for the minor tick
- **Minor tick thickness** : Minor tick thickness.
- **Minor tick height** : Select a height for the minor tick.

Legend

Options to configure the chart legend.



Legend

- **Position** : Selects the position of the legend, “beside” or “Below”..
  - **Beside** : Positions the legend beside the chart.
  - **Below** : Positions the legend below the chart.
- **Allow drag** : If this option is selected it will be possible to drag the chart legend to the desired position.
- **Scroll background color** : Selects the color of the scroll bar.
- **Title font** : Select the font, and your formatting such as bold, italic, size and color of the legend.
- **Background color** : Selects the background color of the legend.
- **Background alpha** : Selects the transparency of the legend.
- **Border Color** : Select the legend border color, and may add a hexadecimal value or select using the color picker..
- **Border width** : Selects the Border width of the legend.
- **Border alpha** : Selects the border transparency of the legend.

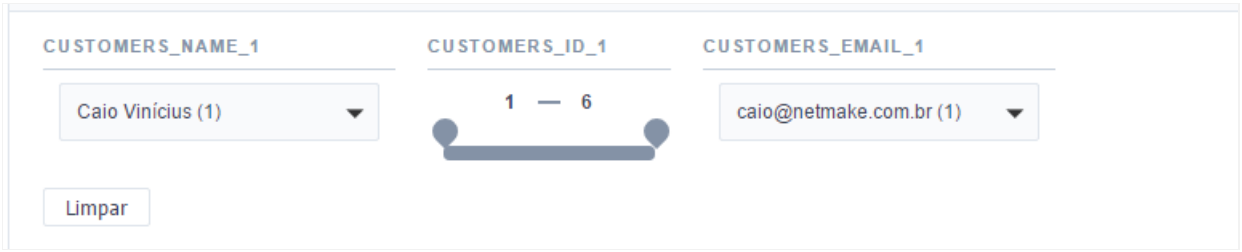
Limit Settings

In the limit option, the user can choose the maximum limit that will be displayed in the chart.

Chart limit Interface

Limit Settings

- **Field** : Field that will realize the filter in the desired limit.
- **Summarization** : In which summarization function will be done in the limit can be “Sum”, “Maximum”, “Minimum”, “Average”, “Variance”, “Standard deviation”, “Count” and “Distinct count”.
- **Limit type** : Defines if the limit will be realized by the firsts or by the last records.
- **Quantity** : Quantity of records that will be applied in the limit.



Generated application

Email Export

It allows you to send the generated export file by email. For that, you may be using SMTP or integration with **Mandrill** and **SES**.

For more information about Mandrill click here

For more information about Amazon SES click here

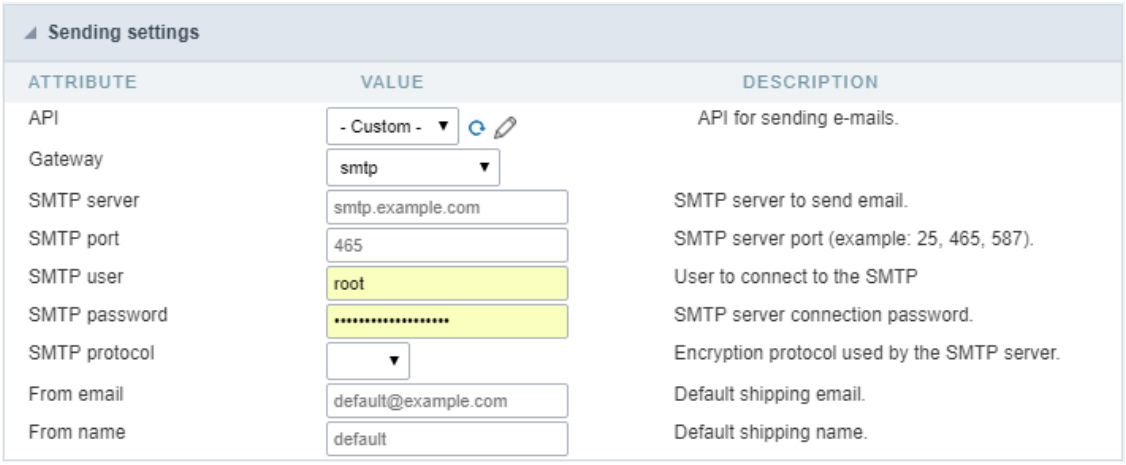
Sending Options

We must configure how to send the email or select an API already set in Tools> API. Click Here and see how to configure.

SMTP

SMTP is the standard protocol for sending emails over the Internet, and each provider has its SMTP.

For more information about SMTP settings click here



- **API**: Allows you to select an API that is already configured in **Tools> API** or set a new one here by selecting **- custom -**. In this case you will see some options according to the selected Gateway.

SMTP Server

SMTP server address for the used provider.

Port SMTP

Define the Port of the mail server. Use the port 465 for security with SSL, 587 for security with TLS, or port 25 for port without security. By default, Scriptcase uses port 25.

User SMTP

SMTP server user.

Enter SMTP

SMTP server user password.

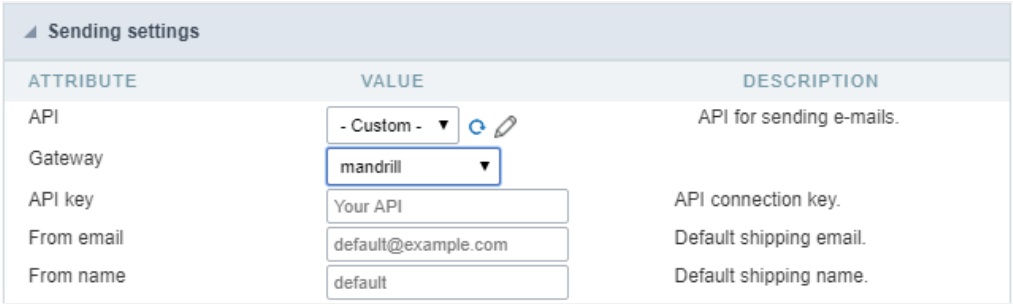
Protocol SMTP

Select the security protocol. If no value is declared, Scriptcase uses the No Security protocol as default.

- **E-mail**: Sets the origin email, which will send the emails.
- **Name**: Name that will be displayed in the sent email.

Mandrill

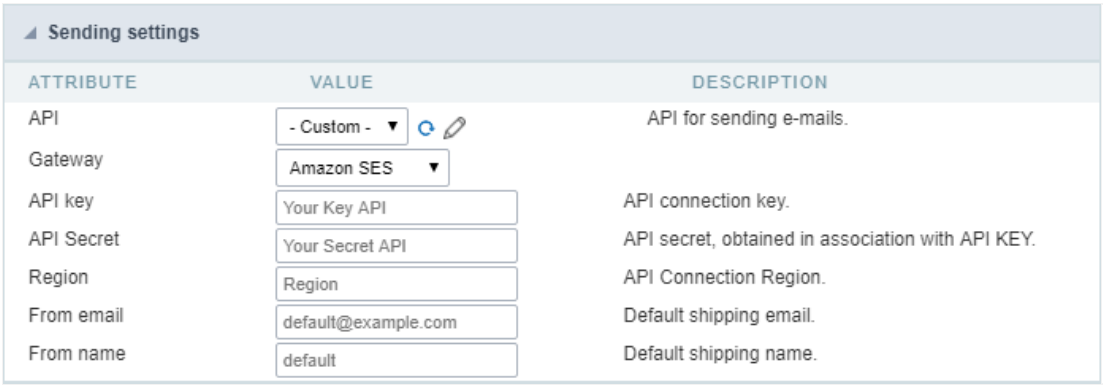
Mandrill is a transactional email API for MailChimp users, ideal for sending data-driven emails.



- **API**: Allows you to select an API that is already configured in **Tools> API** or set a new one here by selecting **- custom -**. In this case you will see some options according to the selected Gateway.
- **API KEY**: Enter the key you obtained when setting up your API.
- **E-mail**: Sets the origin email, which will send the emails.
- **Name**: Name that will be displayed in the sent email.

Amazon SES

Amazon Simple Email Service (Amazon SES) is an email sending service designed to assist in sending marketing emails, notifications, and transactional messages.



- **API**: Allows you to select an API that is already configured in **Tools> API** or set a new one here by selecting **- custom -**. In this case you will see some options according to the selected Gateway.
- **API KEY**: Enter the key you obtained when setting up your API.

API Secret

Enter the secret access key of your Amazon SES account.

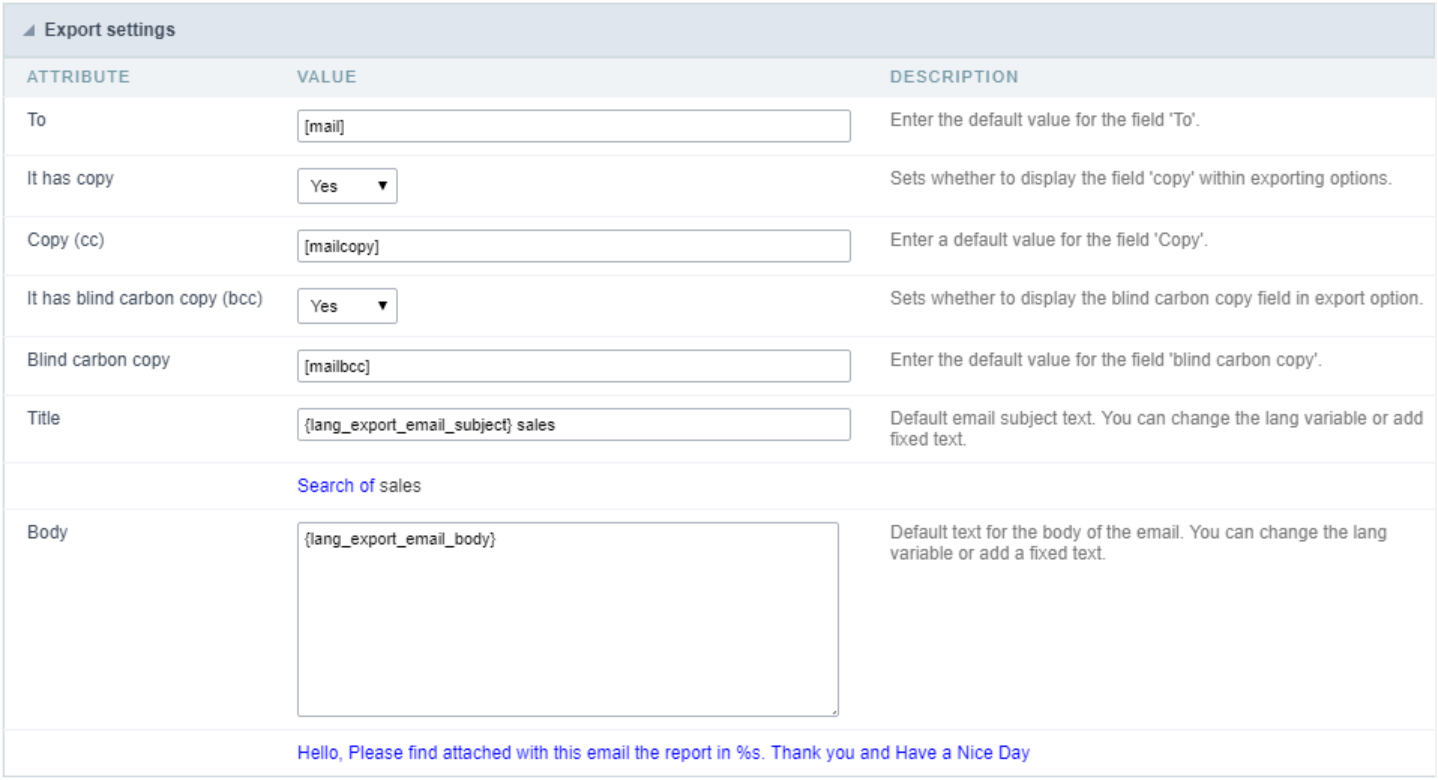
Region

Amazon SES has endpoints in several regions, to reduce network latency, inform the region of the endpoint closest to your application. See the regions.

- **E-mail**: Sets the origin email, which will send the emails.
- **Name**: Name that will be displayed in the sent email.

Export Settings

In these settings, we define the settings of the email sending interface when exporting, and we can define a default email subject and body.



To

Enter the default destination email.

Use Copy (CC)

Defines whether or not to display the 'copy' field within the export options.

- **No**: There will be no possibility to send email with copies.
- **Yes**: The field will be displayed within the export options and we can define a default email for copying. The end user can view this email.
- **Hidden**: The field will not be displayed within export options and we can set a default email for copying.

Copy (CC)

Enter the default email for the copy field. This option can also be empty, or you can use a global variable containing an email list.

Using Hidden Copy (BCC)

Sets whether or not to display the 'hidden copy' field within the export options.

- **No**: There will be no possibility to send email with copies.
- **Yes**: The field will be displayed within the export options and we can define a default email for copying. The end user can view this email.
- **Hidden**: The field will not be displayed within export options and we can set a default email for copying.

Copy (BCC)

Enter the default email for the hidden copy field. This option can also be empty, or you can use a global variable containing an

email list.

Subject

Default email subject text. You can change the lang variable or add static text.

Color

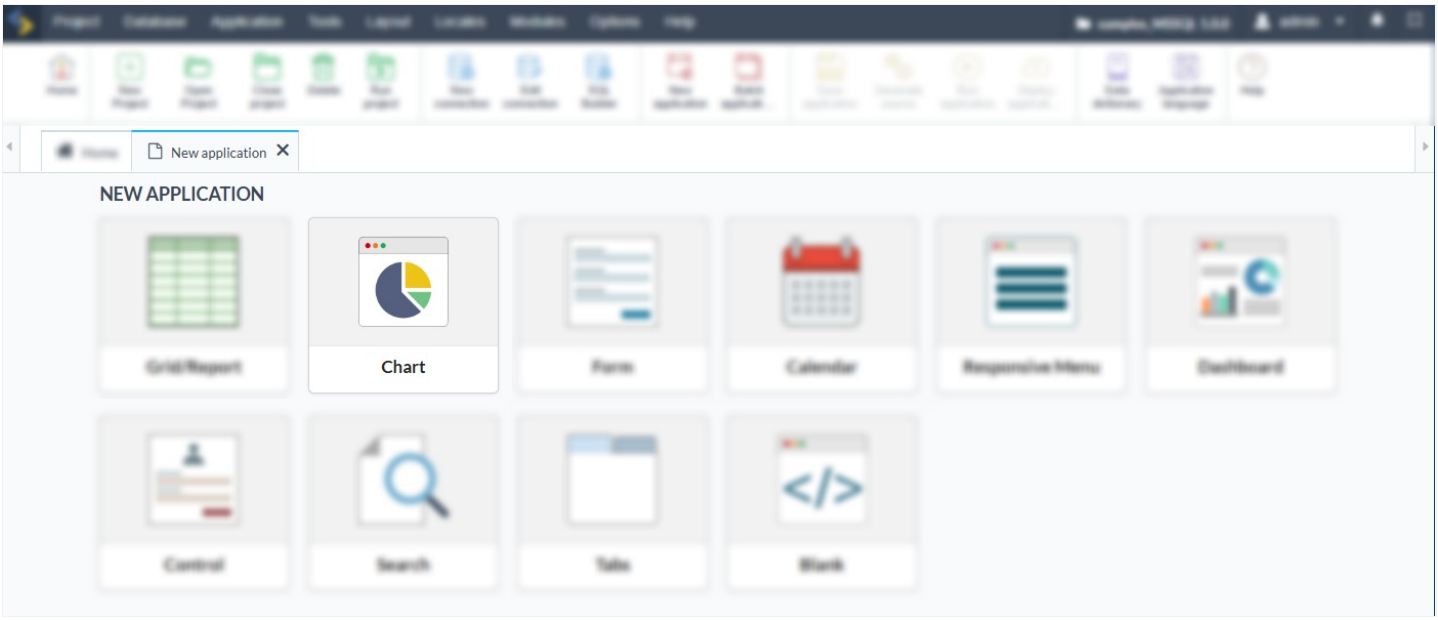
Default body text for the email. You can change the lang variable or add static text.

Creating a Chart Application

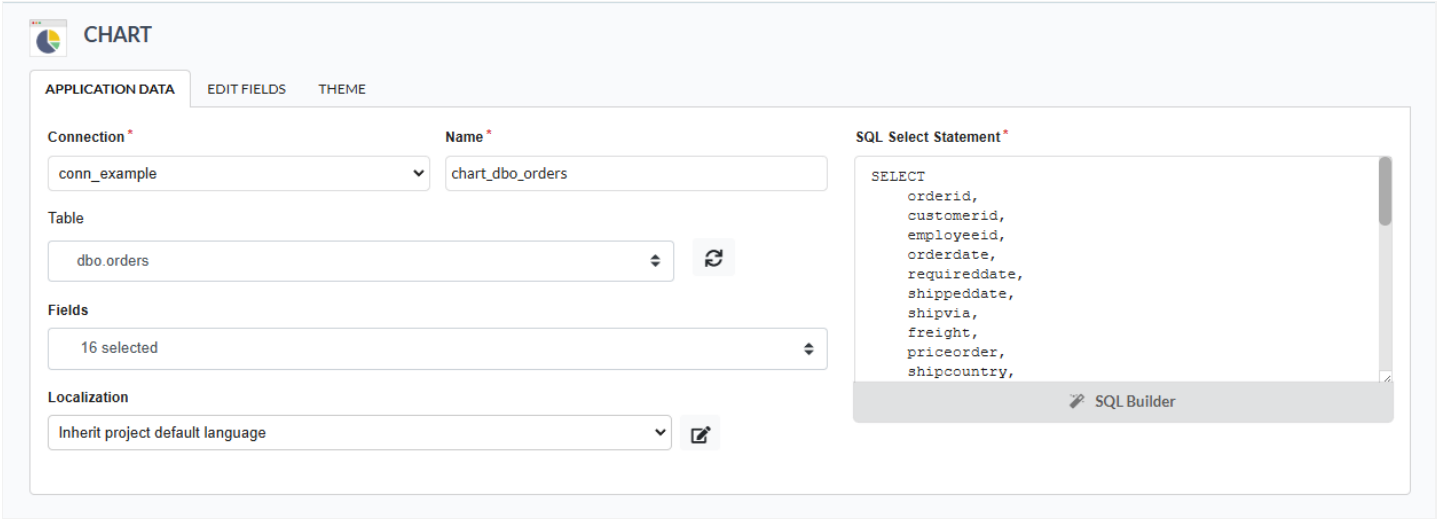
New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.



Application Data



Connection

Defines the connection used for the creation of the application. The existing connections in the project will be listed.

Name

The name of the application being created. It cannot contain special characters.

Table

Defines the tables to be used in the application. (Form and Calendar can only use one table).

Fields

Defines the fields that will be part of the applications.

Localization

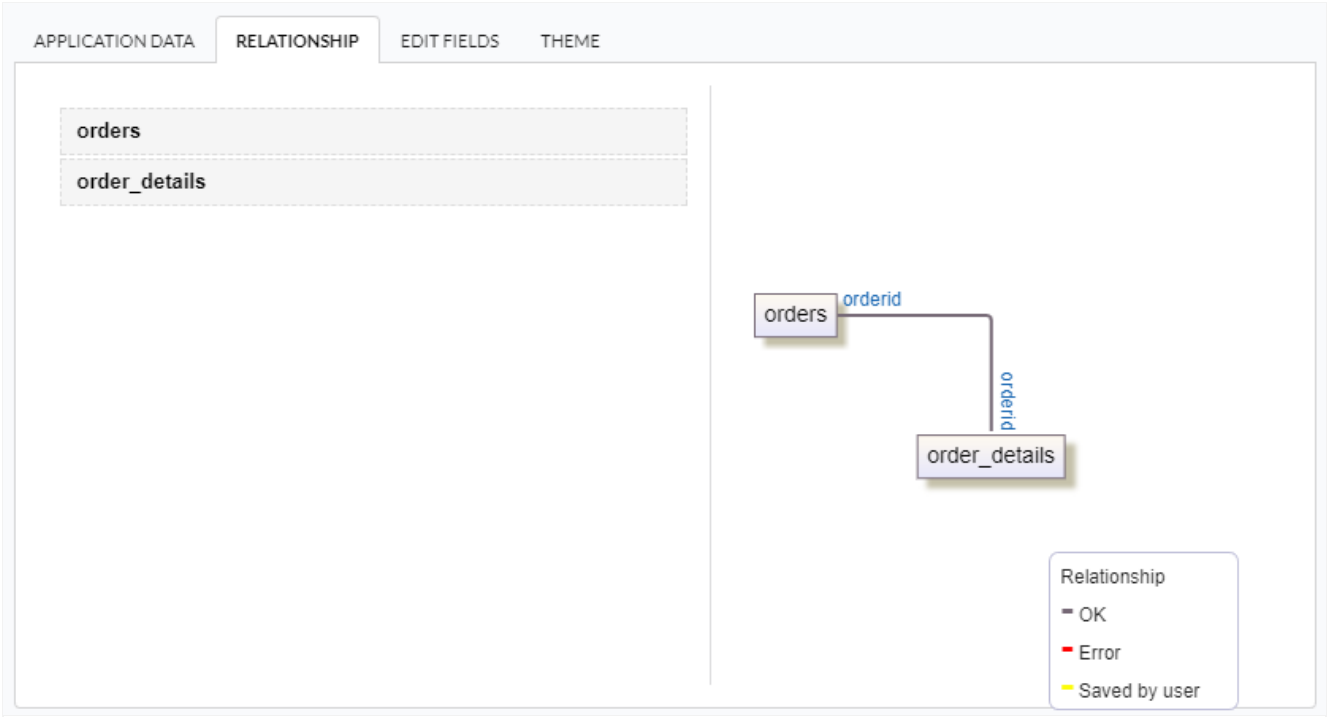
The language of the application to be created. The project's default language is automatically selected.

SQL

Displays the SELECT command generated after selecting the tables and fields. This field also allows the insertion of a previously created SQL statement, as long as it uses tables that exist in the database selected in the connection.

Relationship

When select two or more tables, the tab "Relationship" will be displayed. In this tab we can see the relationship created between the tables, where we can edit the related fields.

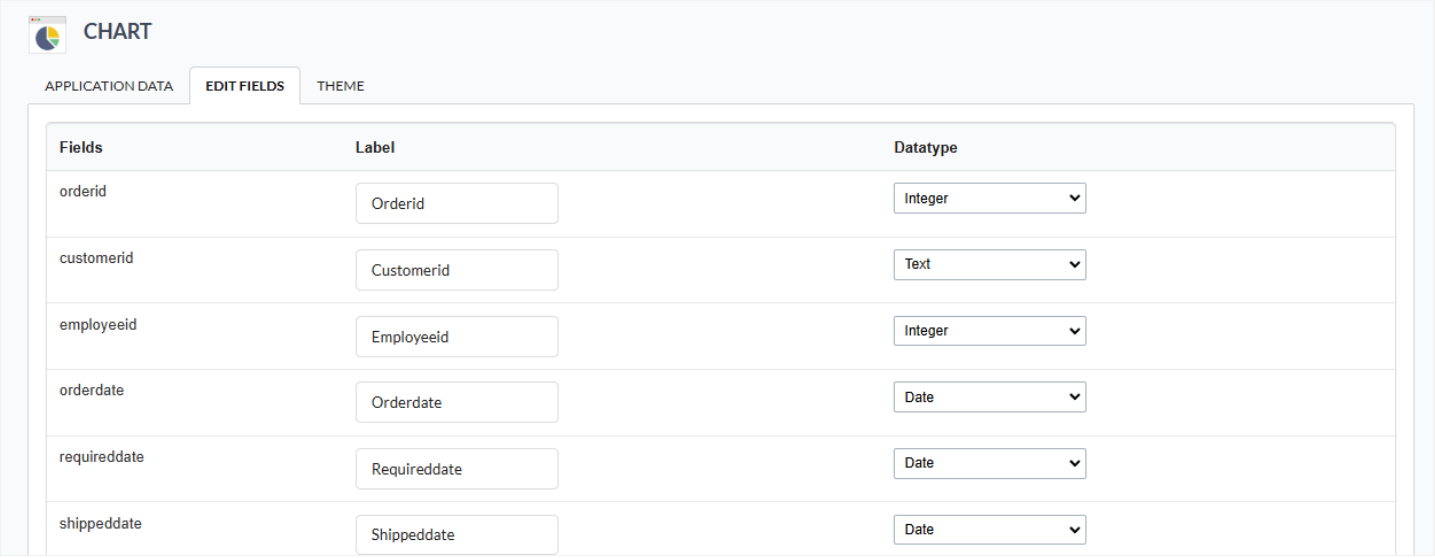


When we click in a link, in the screen above, it will displayed the related field's edition form, as you can see in the image below.

Edit Fields

This screen displays the fields of the selected tables and allows adjustments to be made before creating the application, such as

changing the data type, display name, and other configurations.



Fields

Names of the database fields.

Label

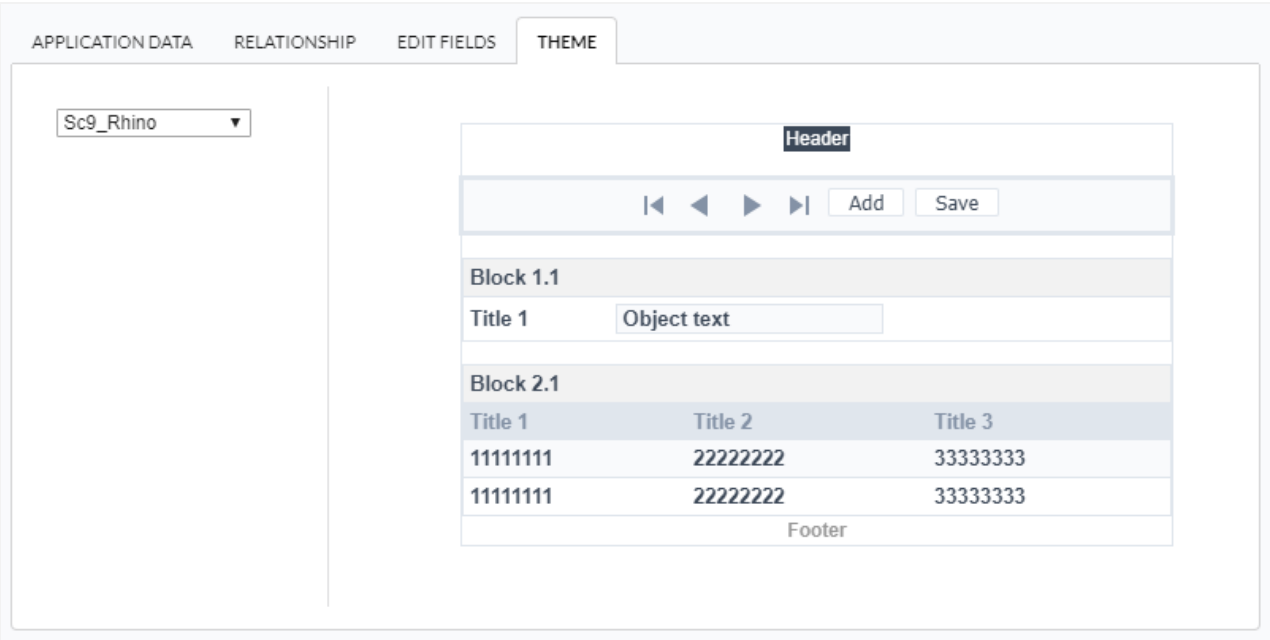
Names of the fields in the generated application's interface.

Datatype

Specifies the field's data type.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.



Related Links

Related Videos ▷

title: Creating a New Grid/Report

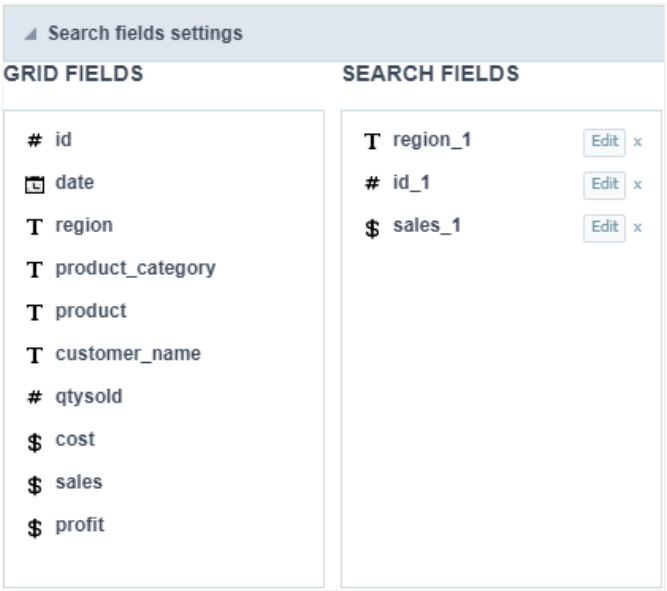
Search

Fields

Search fields settings

To define the fields that will be used in the chart search, use drag-and-drop to position the fields within Search Fields.

The order of the fields, within Search Fields, will be the display order in the generated application.



Editing Fields

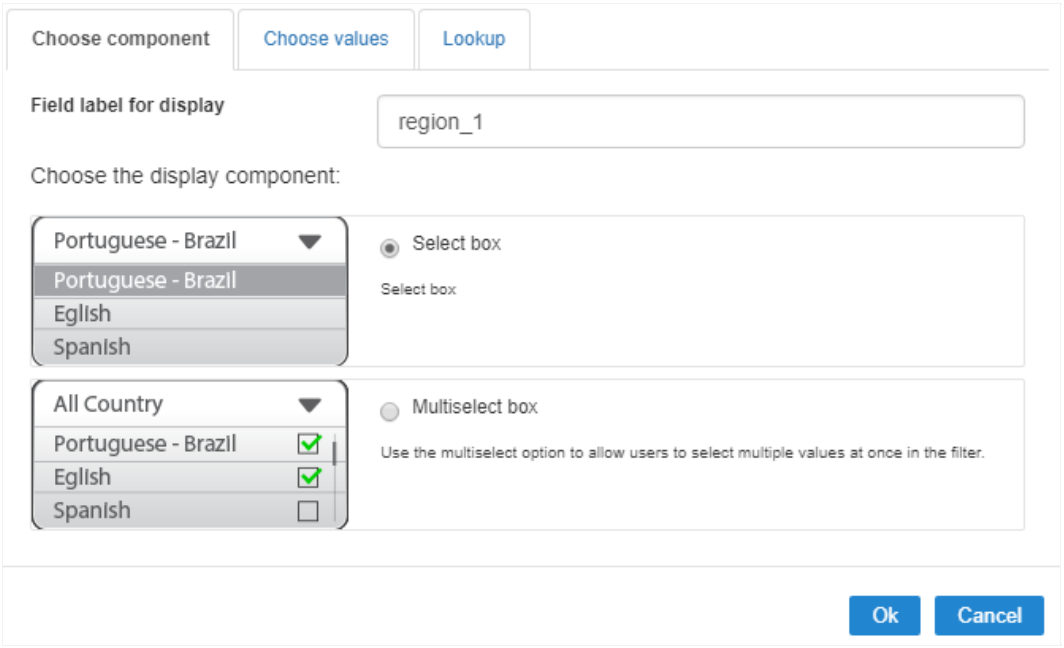
The same field can be added more than once to the Search, these fields will be visible in the generated application since it uses distinct settings, if they are using the same configuration only one of the fields will be displayed.

The display settings of the fields are performed individually, and can be accessed by adding the field in the search and clicking edit.

you will separate the explanation of the display settings from the fields according to the types.

Types: Text/Special

Configuration interface for text and special fields.



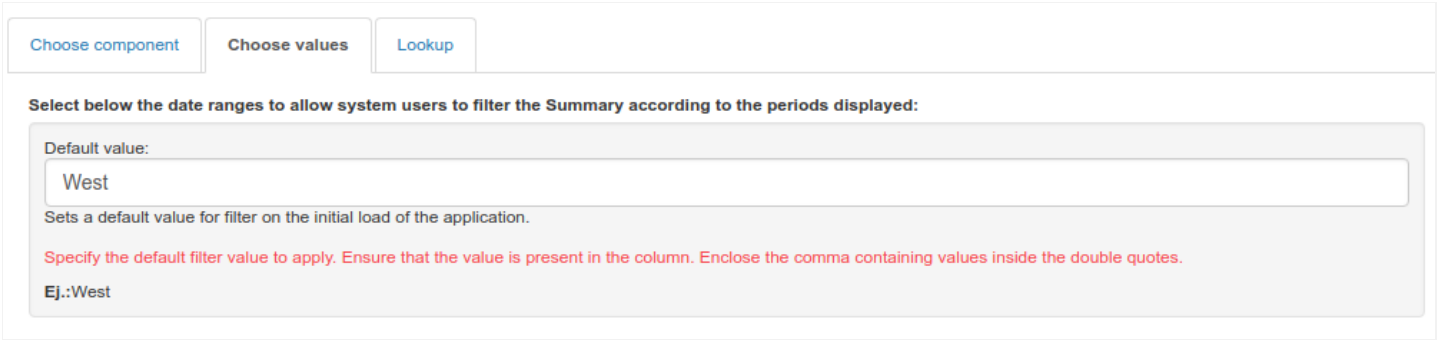
Choose Component

You must define how the field will be used in the Search. Each field type, text, number, and date have different configuration options.

- **Search field label** : Defines the label of the field that will be displayed in the Search.
- **Choose component type**: It defines the format of use of the fields in the Search, for text fields you have the Select Box and the Multi select box.

Choose values

The **default value** field allow us to define a default value to the chart search in the initial application charge.

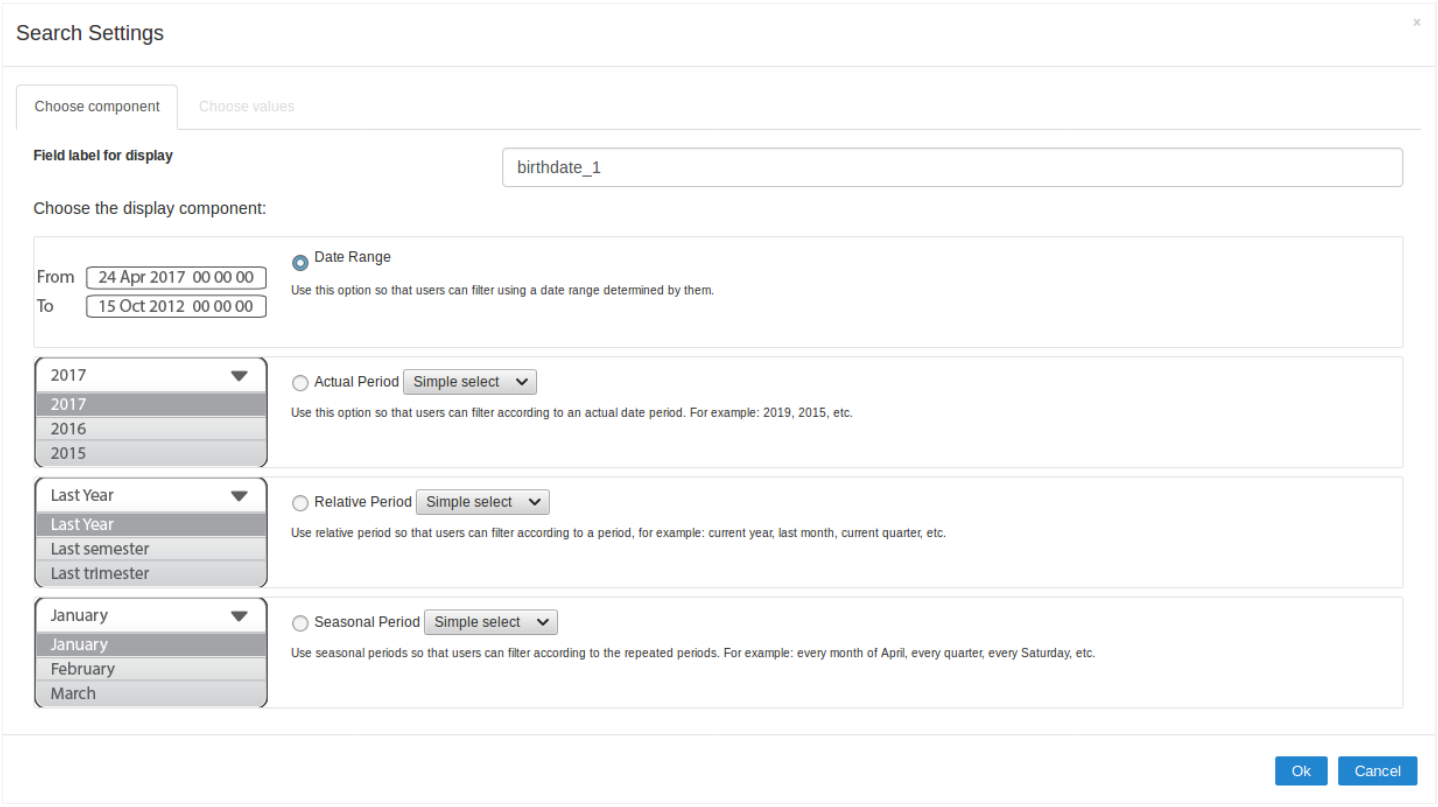


Lookup

In this tab, you can configure the display lookup in the Search field. For more information on creating a view lookup, access **Application > Grid > Fields > Text**.

Types: Date/Time

Configuration interface for date and date time fields.



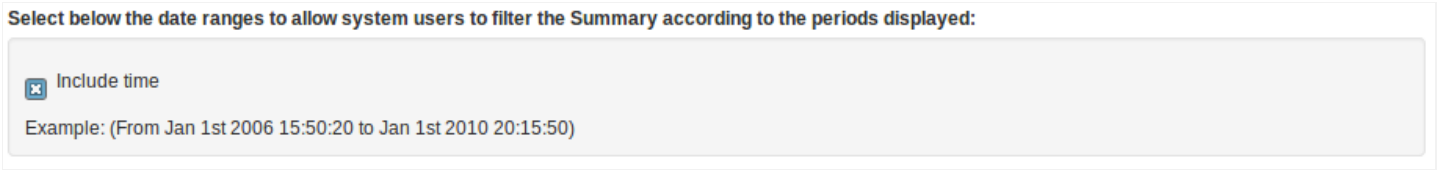
Choose Component

You must define how the field will be used in the Search. Each field type, text, number, and date have different configuration options.

- **Search field label** : Defines the label of the field that will be displayed in the Search.
- **Choose component type**: Defines the format of use of the fields in the Search, for fields date and datetime, you will have the following options: Date Range, Actual Period, Relative Period and Seasonal Period.

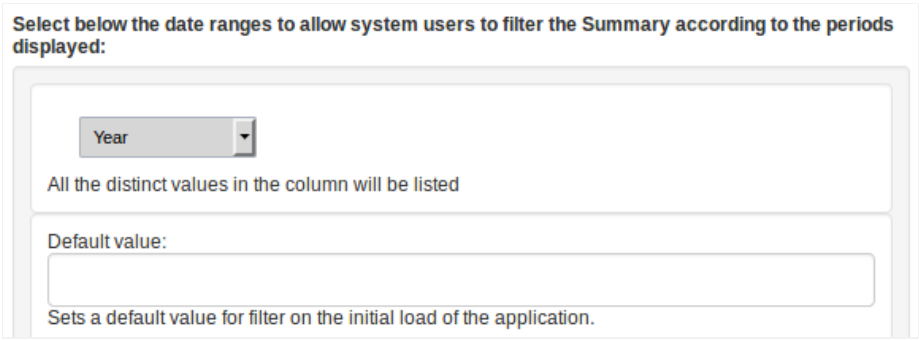
Choose Values - Date Range

For datetime fields, you must enable the option **Include Time** so that the hours can be included in the search



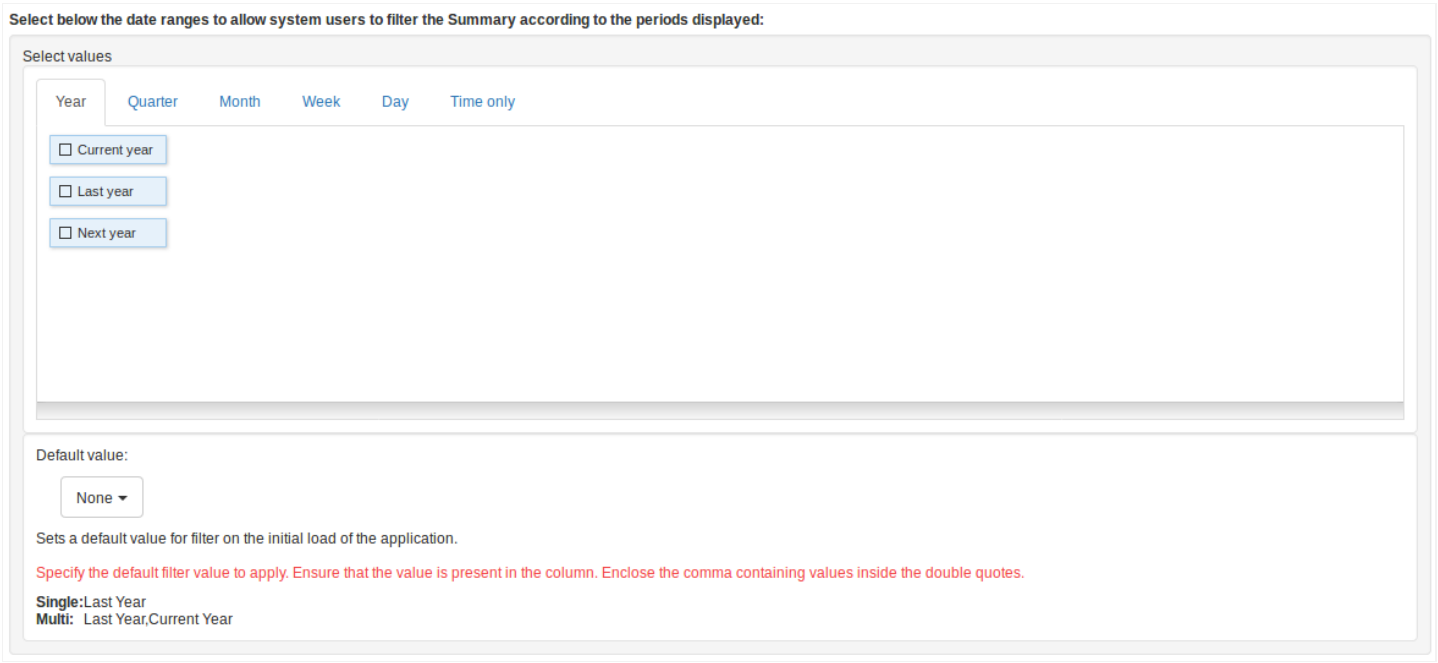
Choose Values - Actual Period

You must define the period that will be used in the search. When running the application, you will have a select with the dates displayed according to the selected period.



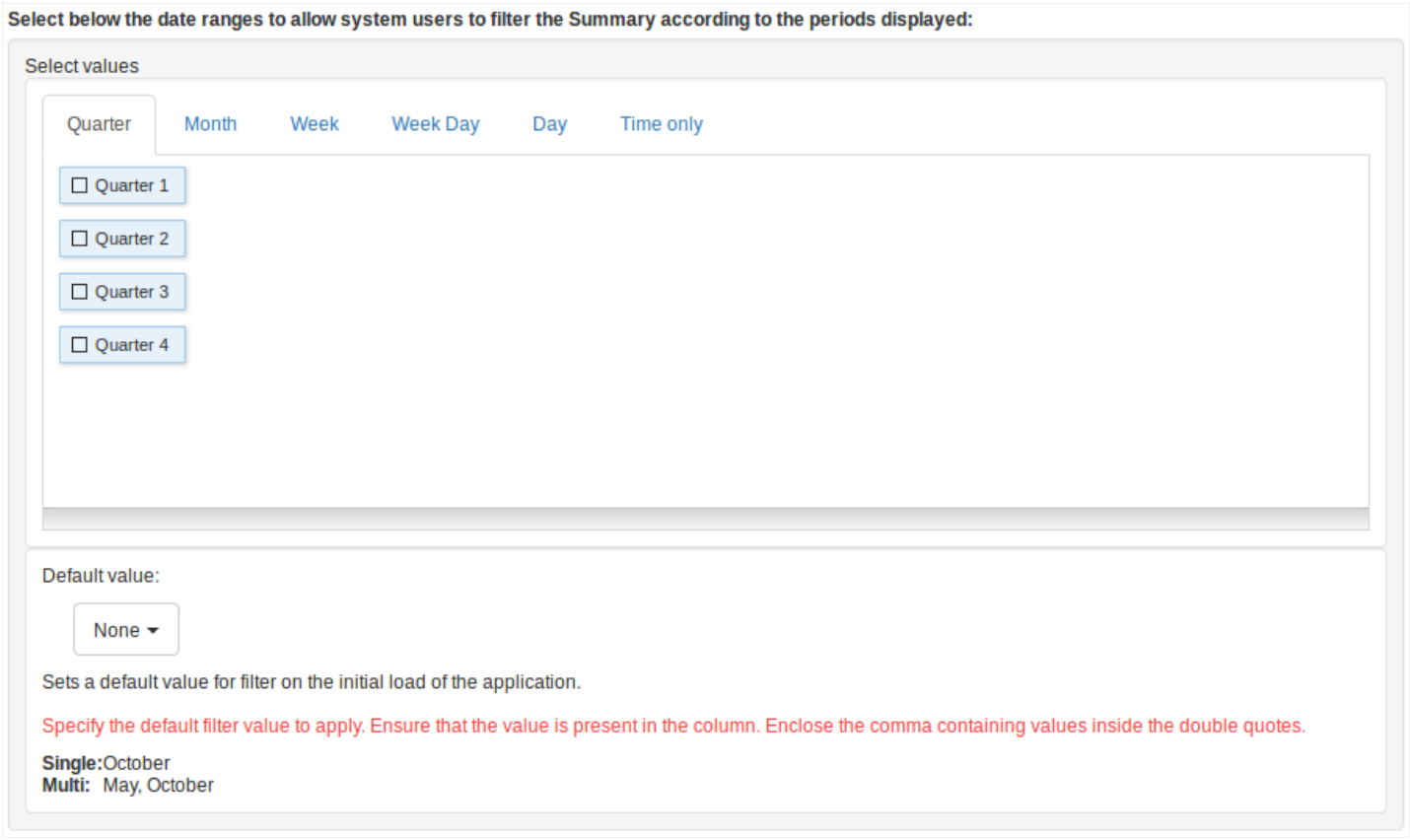
Choose Values - Relative Period

You must define which periods are available for use in the search.



Choosing Values - Seasonal Period

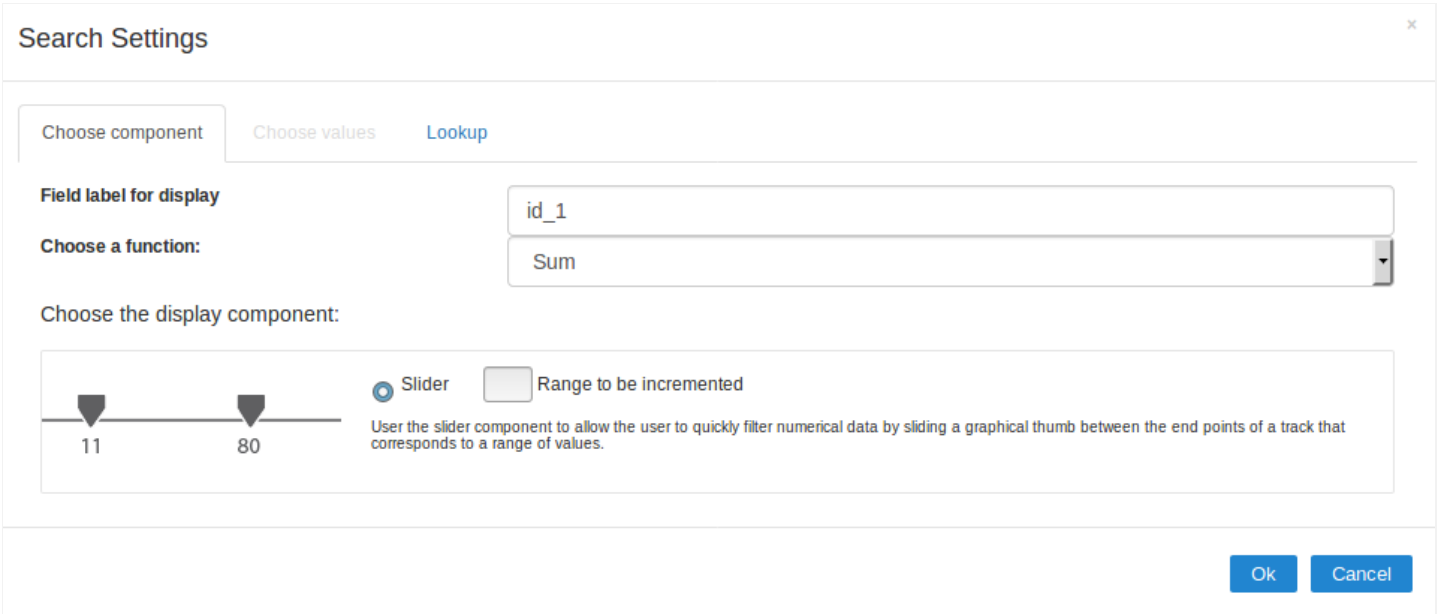
The available values are separated by tabs:



- **Quarter**: Defines the quarters used in the search.
- **Month**: Defines the usage of the months of the year.
- **Week**: Defines the usage of the weeks of the year.
- **Week Day**: Defines the use of the days of the week.
- **Day**: Defines the usage of the days of the month.
- **Only Time**: Defines the use of the day time (Available only in the datetime field)

Types: Number

Configuration interface for numeric fields.



Choose Component

You must define how the field will be used in the Search. Each field type, text, number, and date have different configuration options.

- **Search field label** : Defines the label of the field that will be displayed in the Search.
- **Choose function to apply**: Defines the function that will be used in the search for setting the value. For example, when choosing sum, the range is the range between the lowest and the result of the sum greater.
    - **Actual Values**: This option sets the display of the actual value of the field, saved in the database, without the use of any of the aggregate functions. When using this option, two types of searches, select box and Multi select box usage are added, as well as enabling lookup settings for numeric fields.
- **Choose component type**: Defines the format used for the fields in the search, for numeric fields, you have the Range and selecting **Actual Values** will be displayed **Select Box** and **Multi select box**.

Lookup

For numeric fields, this option is available only when you use the **Actual Values** option and the way the field is used in the search is **Select box** or **Multi select box**.

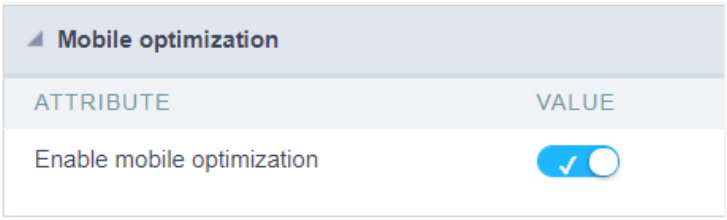
In this tab, you can create a display lookup in the search field. For more information about creating view lookup, go to **Application > Grid > Fields > Integer**.

Chart Mobile Optimization

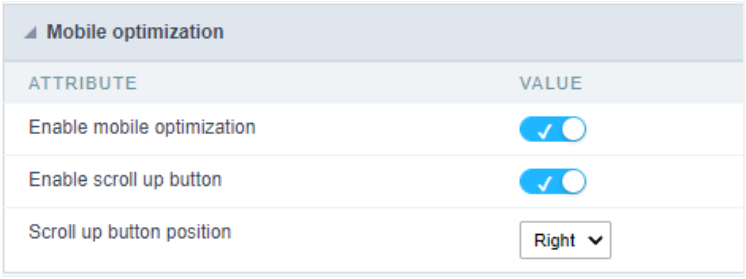
The Mobile settings allows to define the automatic optimization of generated applications to run on mobile devices.

See below the available settings.

Mobile optimization



Enable mobile optimization

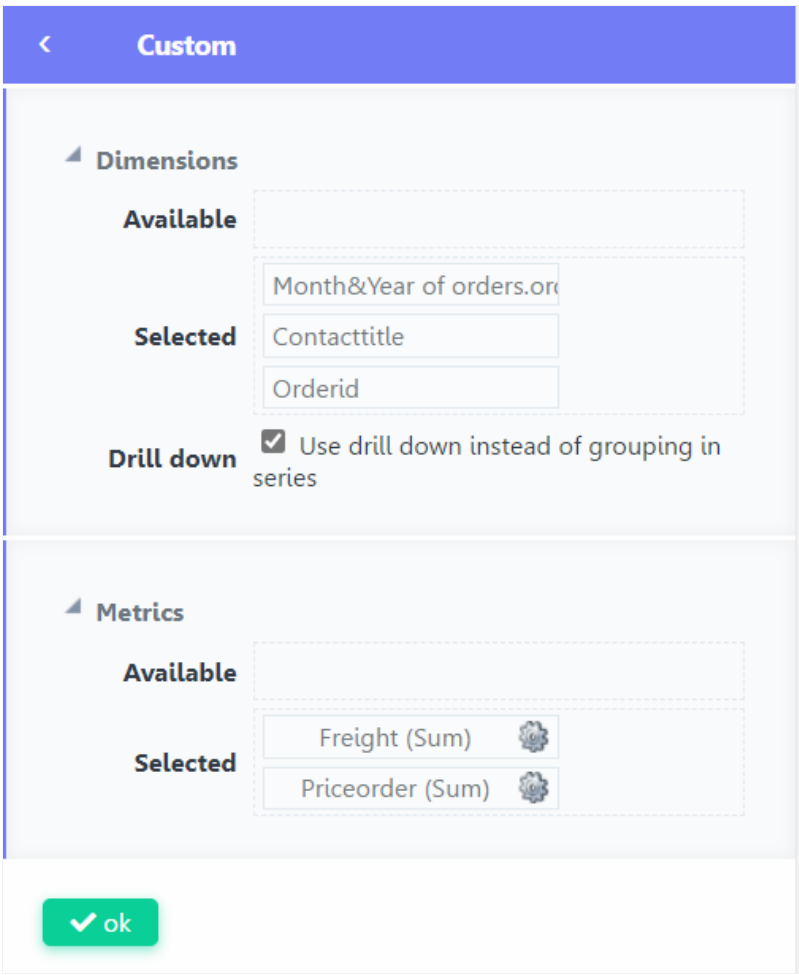


This option changes how the application HTML elements works, adapting them automatically to run on mobile devices.

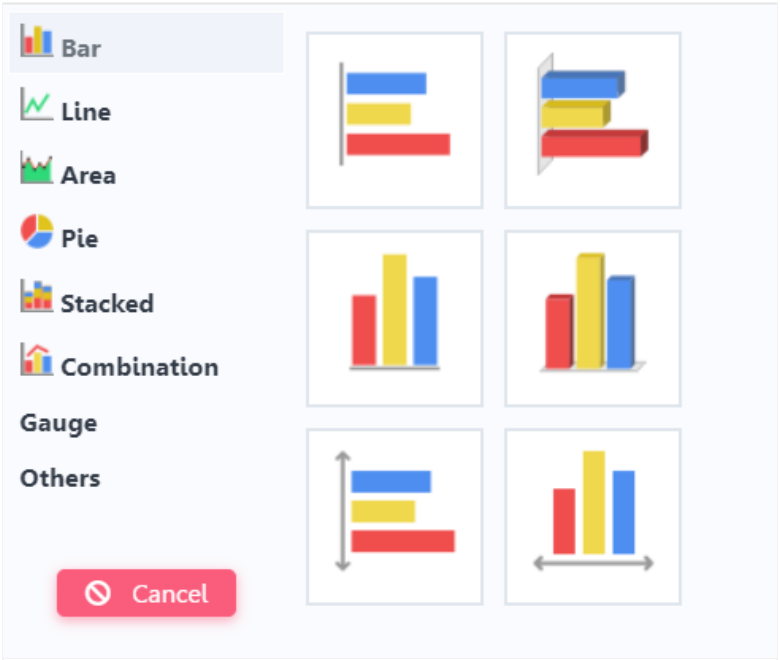
It enables this option by default when creating a new application.

See some examples of adapted screen:

Example of graphic setup screen on mobile device



Example of chart type selection screen on mobile device

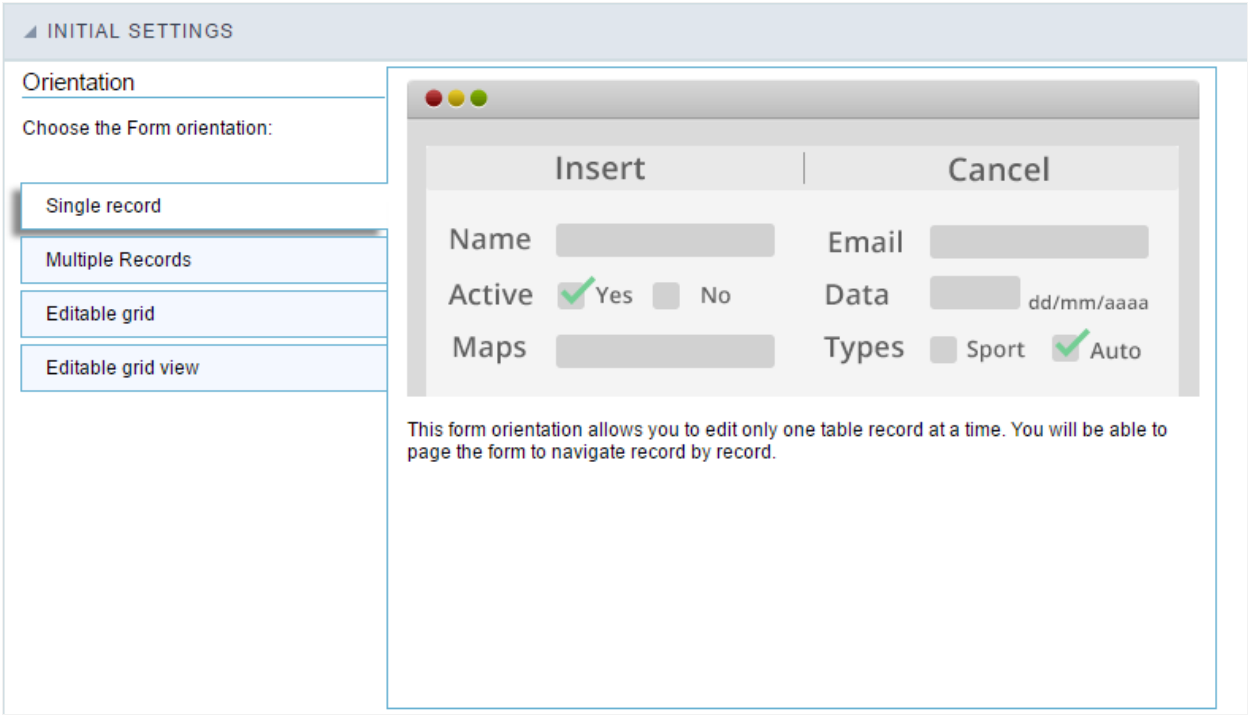


Form Orientation

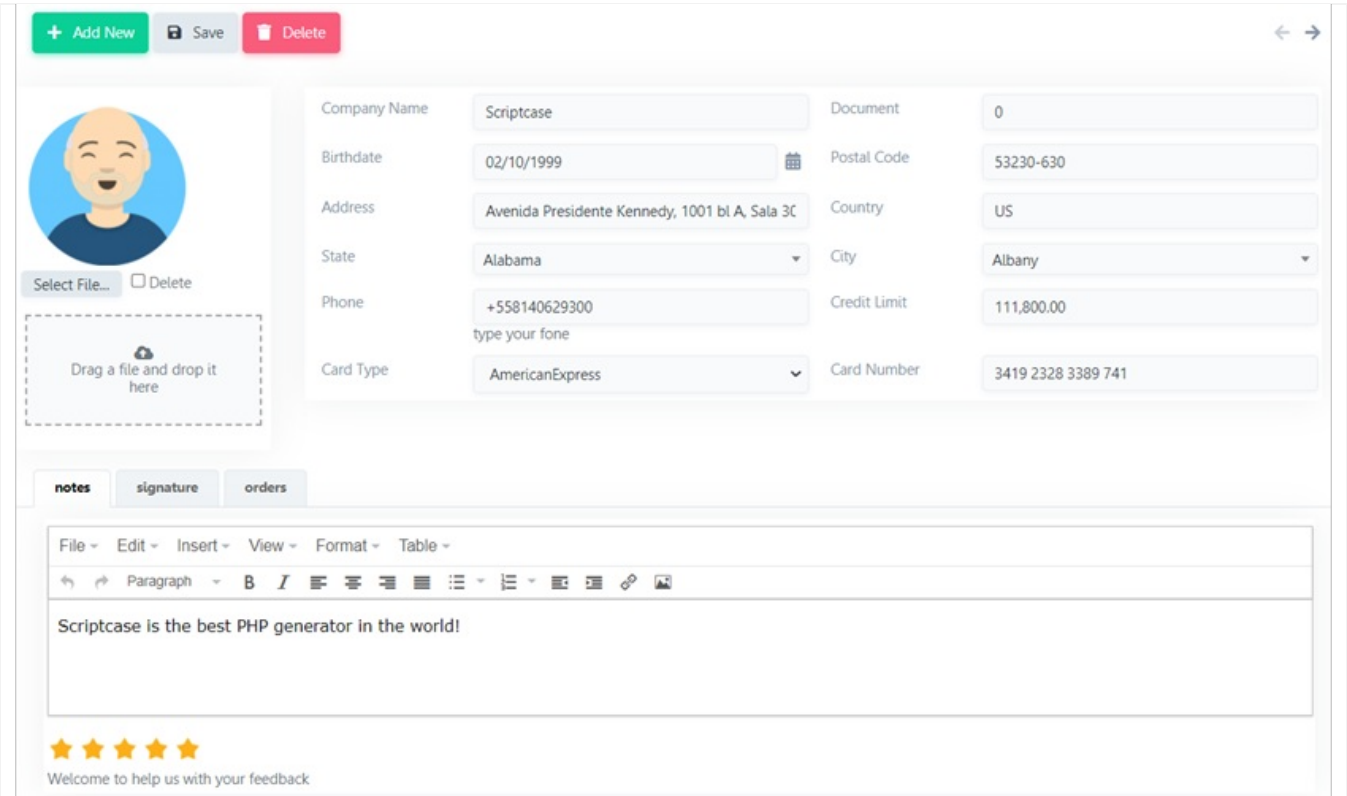
Orientation models for the Form application can be Single record, Multiple records (batch editing), Editable grid (inline editing), and Editable grid view.

Single record

This form orientation allows you to edit only one table record at a time. You will be able to page the form to navigate record by record.



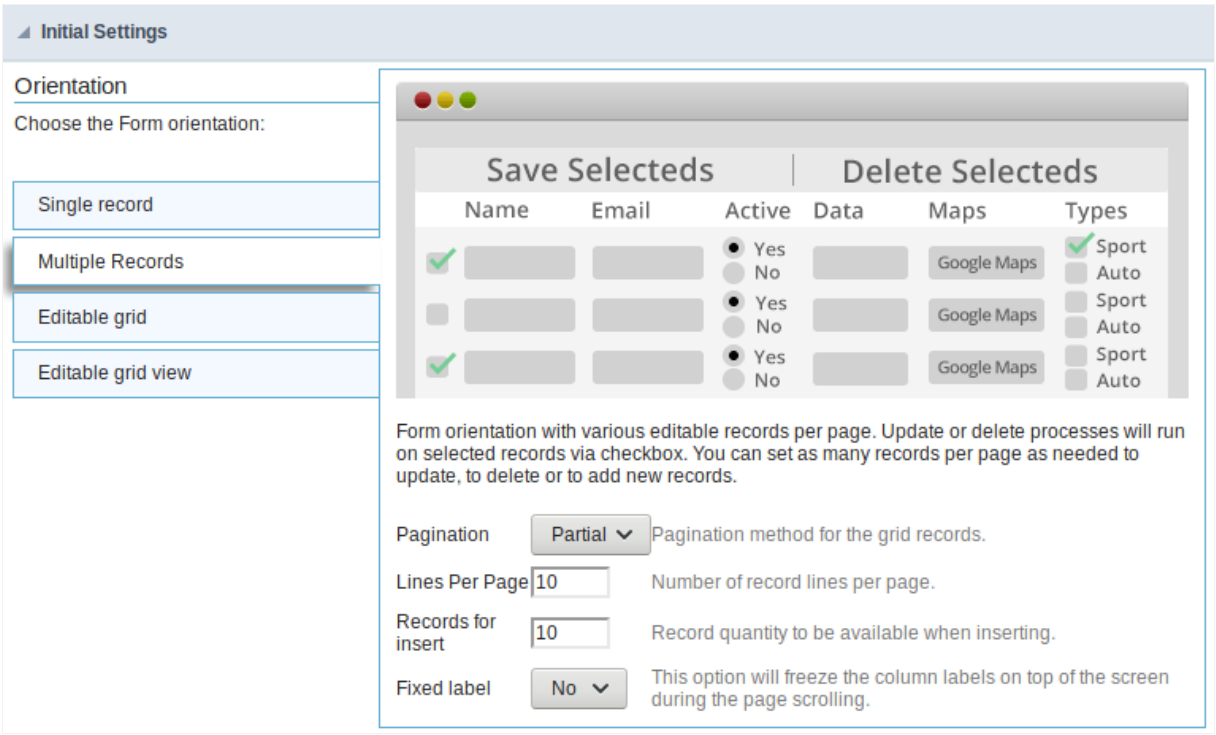
See below a single record form example; you will have different options to organize the layout by adding the fields in different positions using columns, tabs, blocks, and pages. Check the Form layout options for more details.



Single record form - application example

Multiple records (Batch editing)

Form orientation with various editable records per page. Update or delete processes will run on selected records via check box. You can set as many records per page as needed to update, delete or add new records.

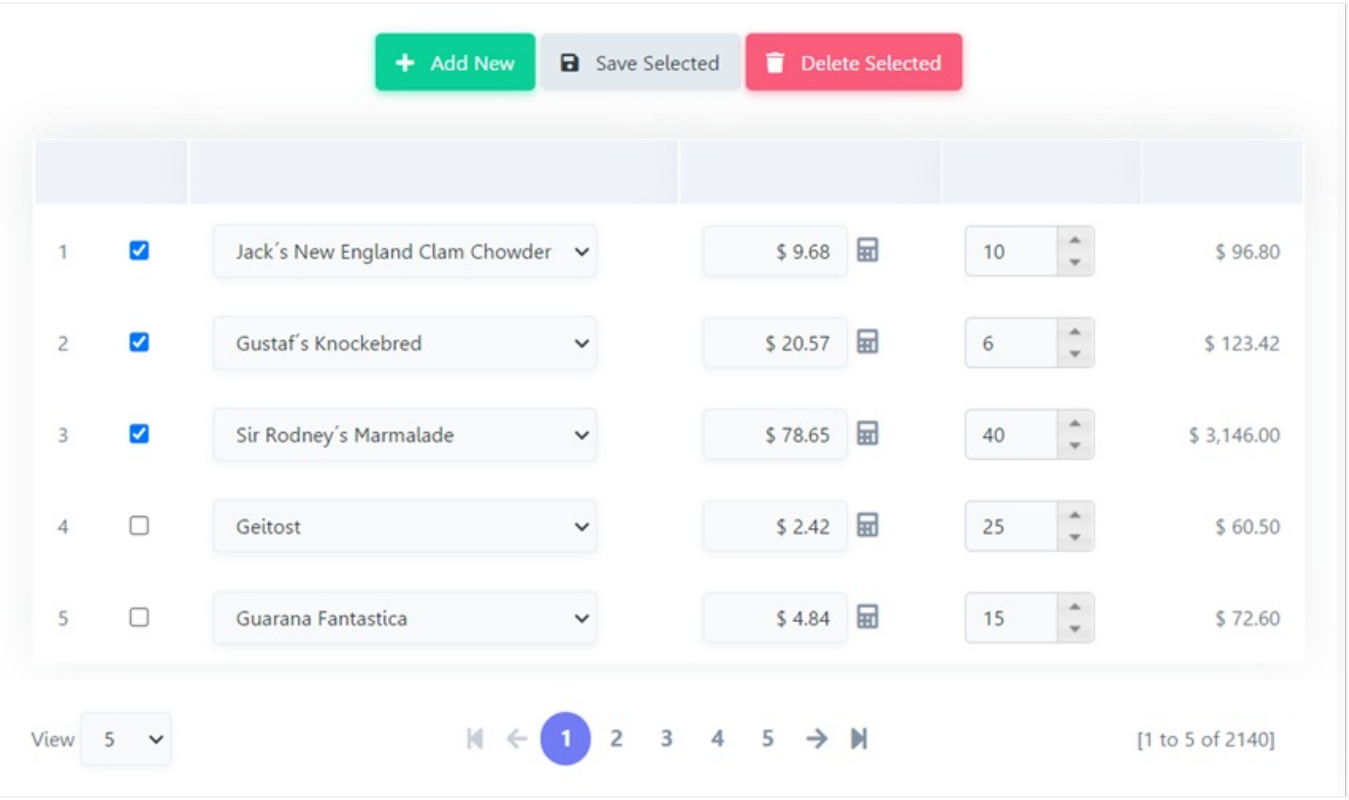


- **Pagination** : Paging type for the multiple records form. You can choose if it will be partial or total.

*> **Partial pagination** divides the records according to the number of pages defined inside the "lines per page" option. You can also add a button "View" within the application to change dynamically the number of records displayed per page. See here how to change the amount of records options; **Total pagination** displays all records in just one page.*

- **Lines Per Page** : Quantity of rows of records per page. This option is available only when the pagination is “Partial.”
- **Records for insert** : You can set here the number of rows of records per page in the insert mode to add multiple records at a time.
- **Fixed Label**: Establishes the column label at the screen top while scrolling the page.

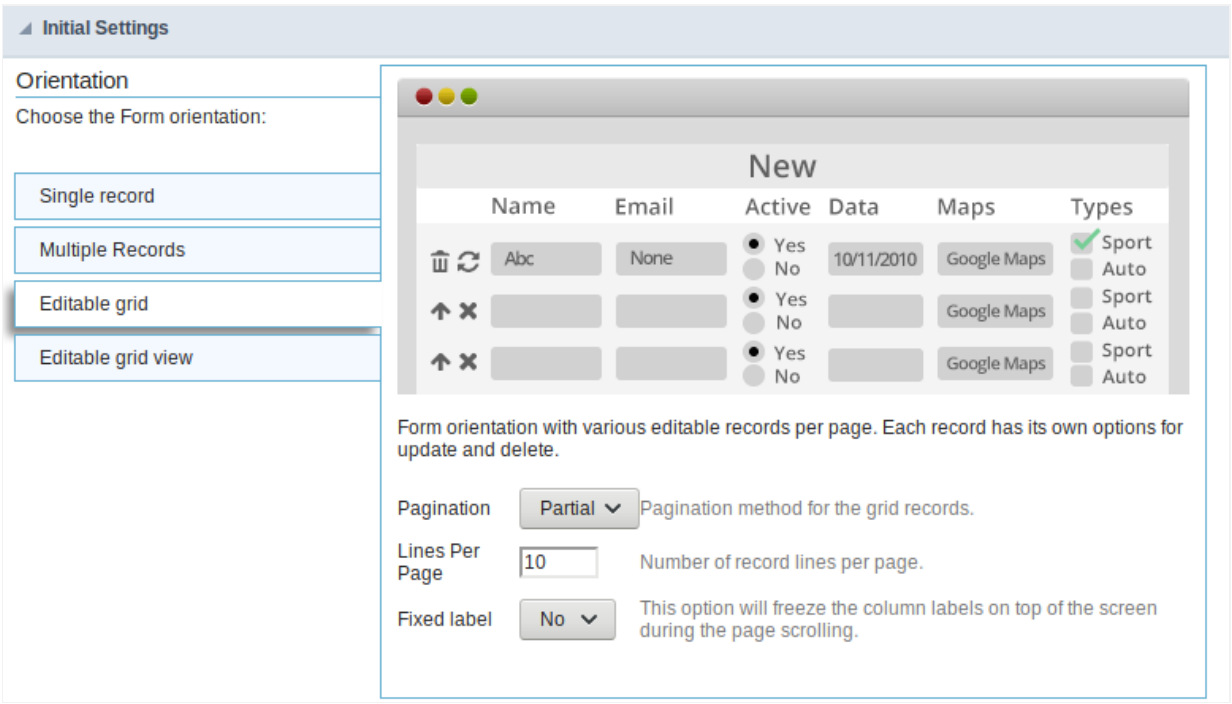
In the example below, you can see one multiple records form type displaying five records per page for edition.



Multiple records form - application example

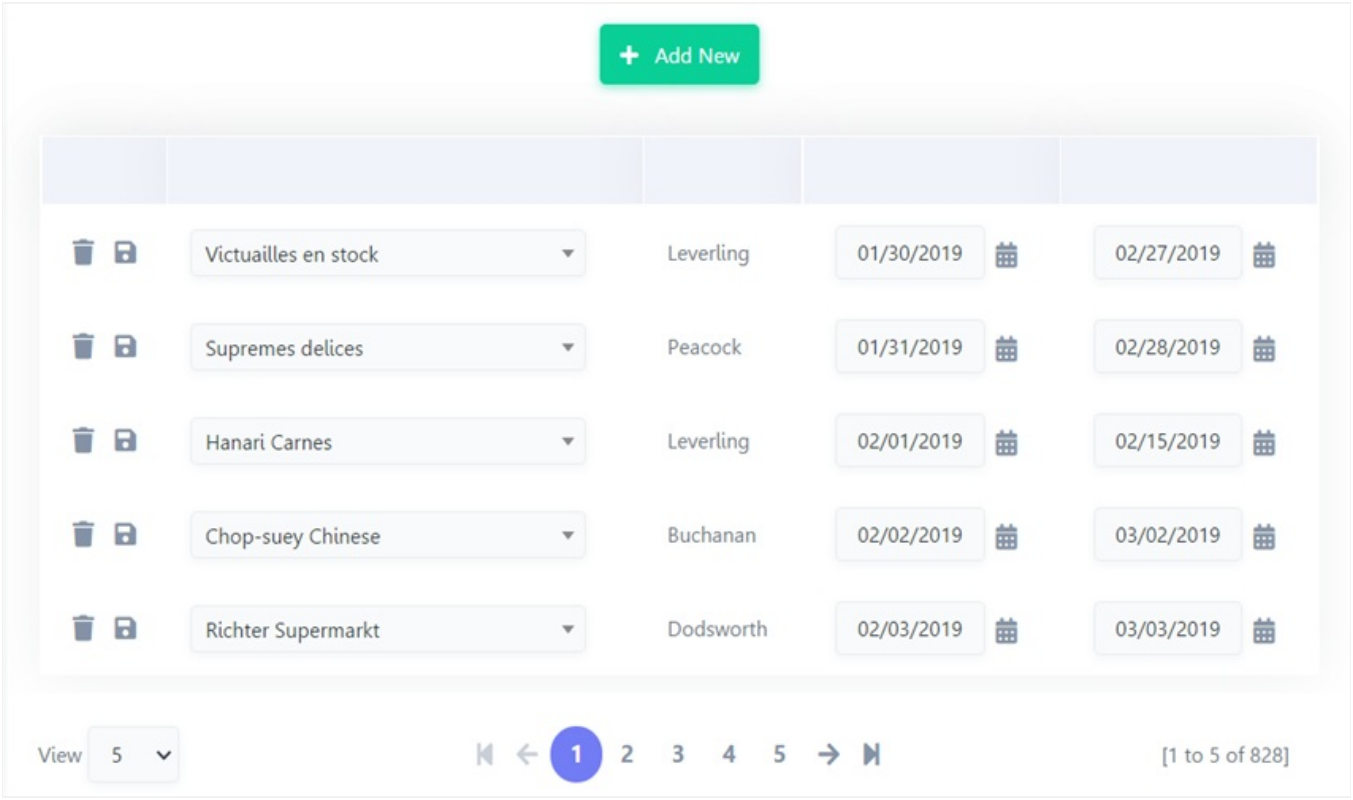
Editable grid (inline editing)

Form orientation with various editable records per page. Each record has the update and deletes controls available in its line.



- **Pagination** : Paging type for the multiple records form. You can choose if it is going to be partial or total.
- **Lines Per Page** : Quantity of rows of records per page. This option is available only when the pagination is “Partial.”
- **Fixed Label**: Establishes the column label at the screen top while scrolling the page.

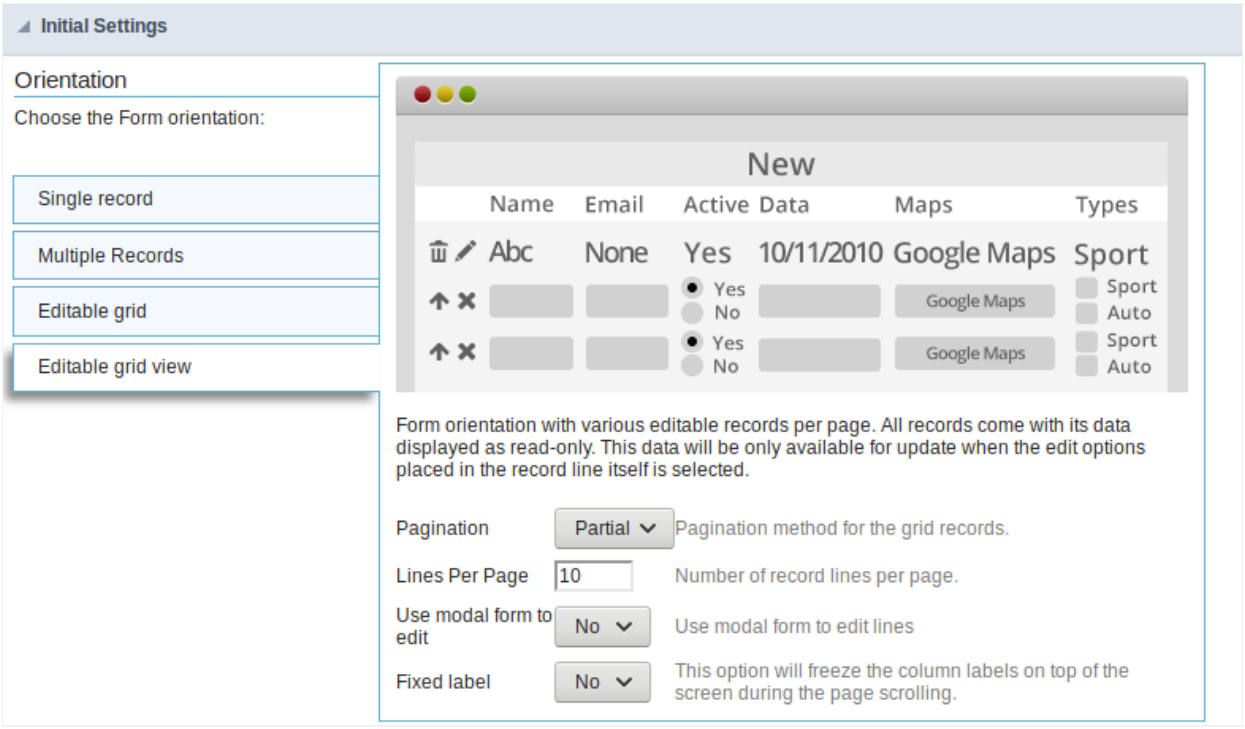
As you can see in the example below, using the Editable grid, you will have the form with multiple records, and you can edit more than one at the same time. However, differently from the multiple record, you will have to add line by line.



Editable grid form - application example

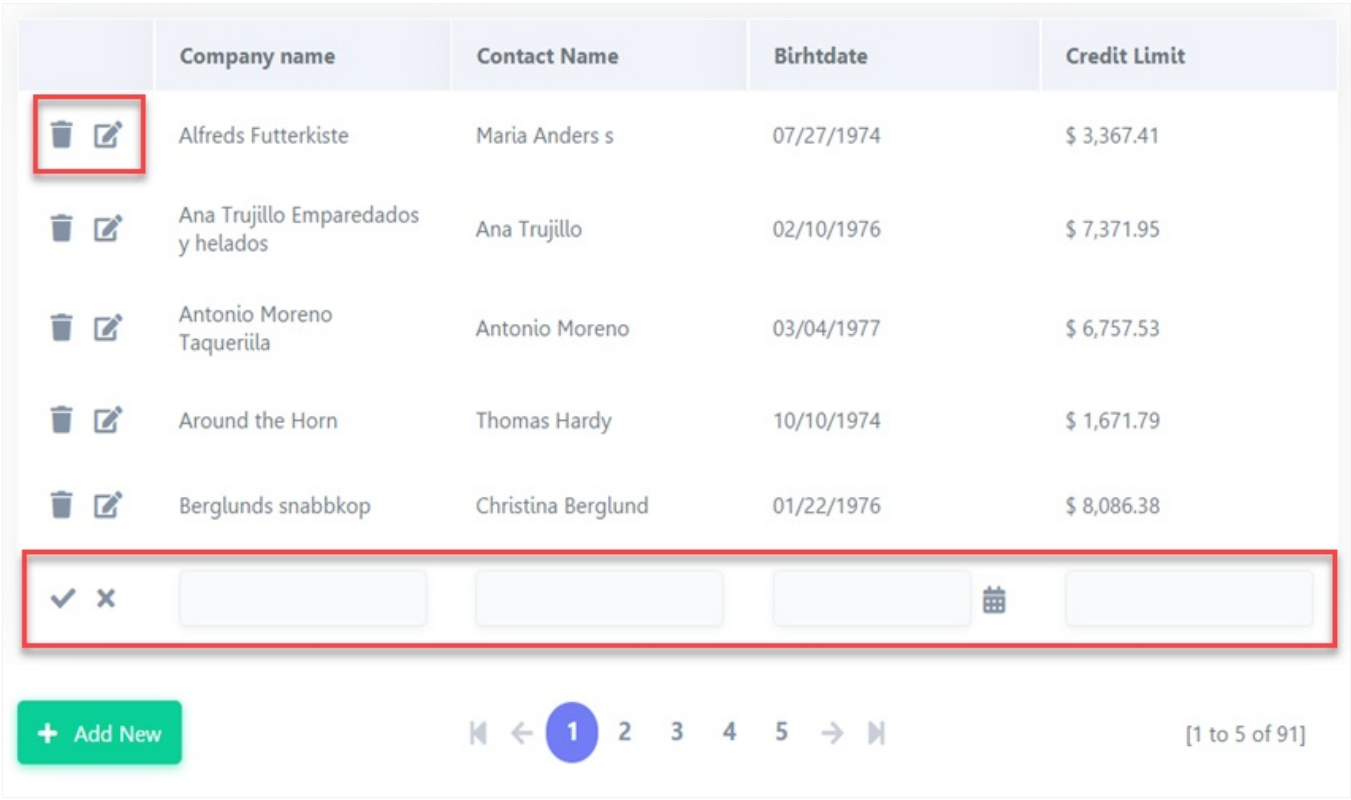
Editable grid view

Form orientation with various editable records per page. All records come with their data displayed in a read-only format. These data are only available for the update by selecting the Edit option in the record itself. You will have the opportunity to edit the record inline or using a modal.

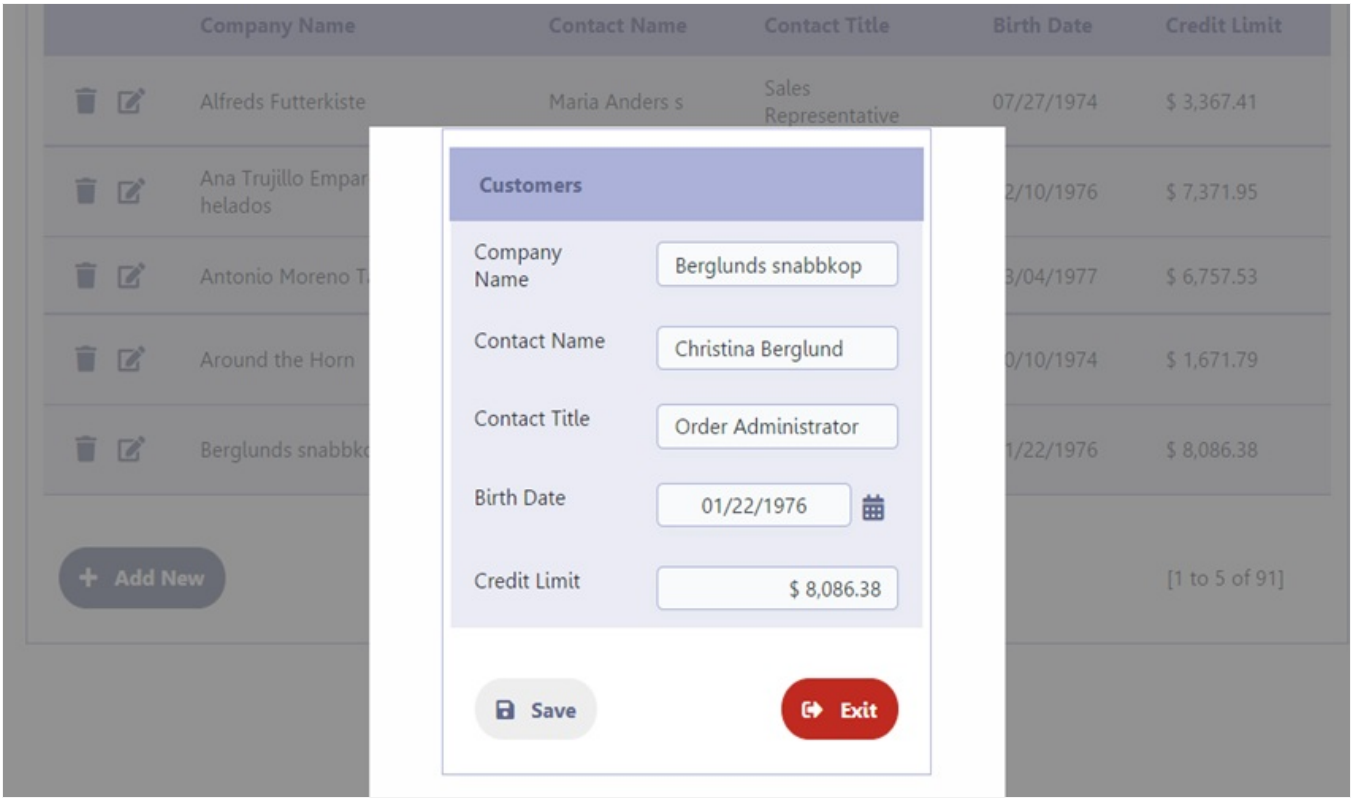


- **Pagination** : Paging type for the multiple records form. You can choose if it is going to be partial or total.
- **Lines Per Page** : Quantity of rows of records per page. This option is available only when the pagination is “Partial.”
- **Use modal form to edit** : Use a modal form to edit the records when you click to edit the record.
- **Fixed Label**: Establishes the column label at the screen top while scrolling the page.

The editable grid view form example below shows you you will edit a record. You can’t edit multiple records. You will have the grid view and the options to manipulate one by one using the inline edition option or opening a modal.



Editable grid view form - in line edition example



Editable grid view form - modal edition example

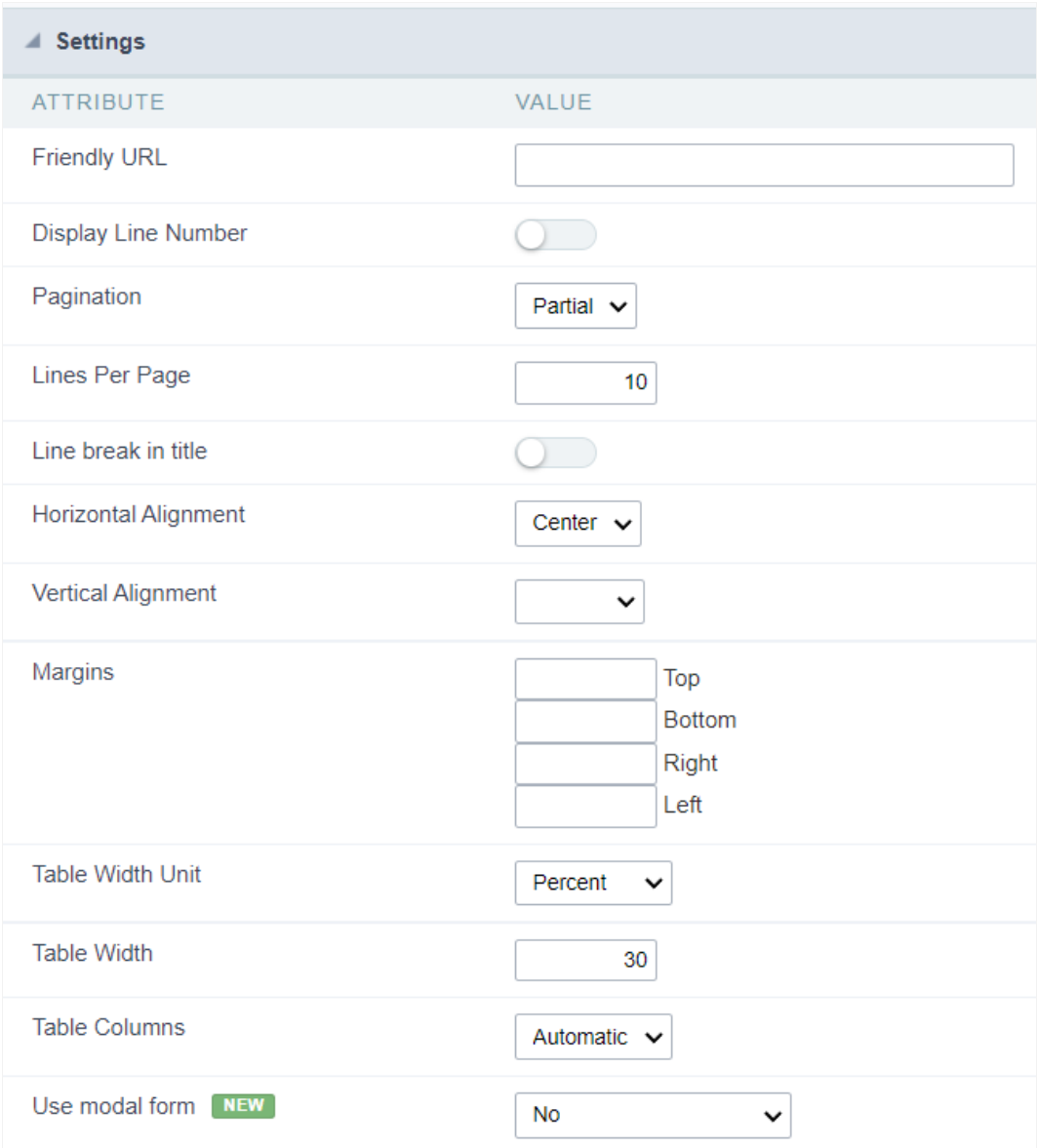
Related Links 🔗

Related Videos ▷

Form Settings

General settings of the menu form where it is possible to configure some behavior of the application depending on the orientation used.

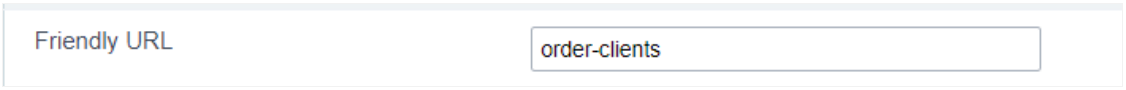
Settings



Check the details of each attribute available in the configuration block below.

Friendly URL

This attribute defines the application's friendly URL.



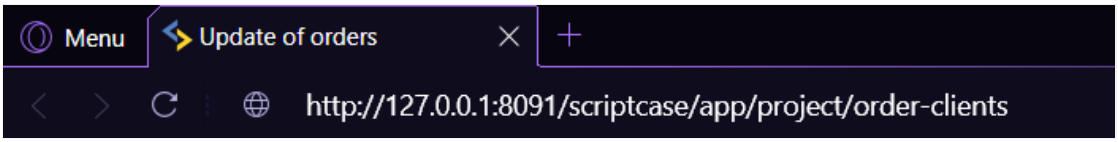
Alphanumeric characters and some special characters are allowed, such as: **hyphen** (-), **underscore** (_), **comma** (,), and **dot** (.).

The use of accents or spaces is not allowed.

Some recommendations for definition are:

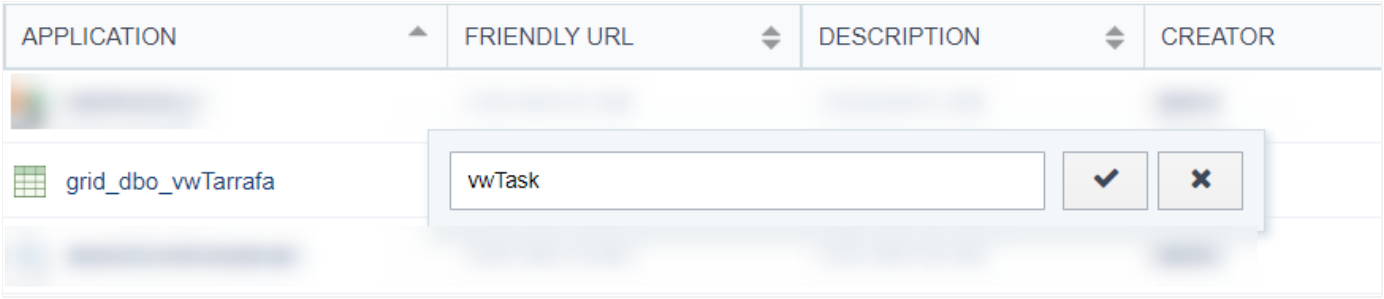
- Use keywords for identification.
- Use hyphens to separate words.
- Use only lowercase letters.
- Avoid using dates.

Example of an application using a Friendly URL



The friendly URL can also be defined in the application list on the project's initial screen.

See the example below

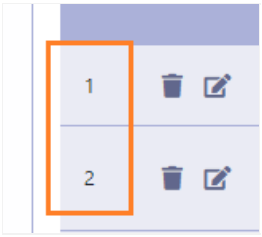


Display Line Number

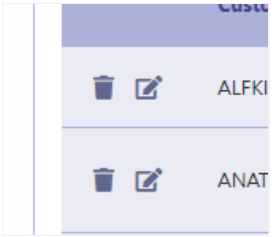
This attribute defines whether or not the line counter is displayed in the application. The row counter will be positioned next to the options column.

This feature is only available for forms configured in the formats: **Grid Editable** and **Grid Editable (View)**

Example with display line number enabled

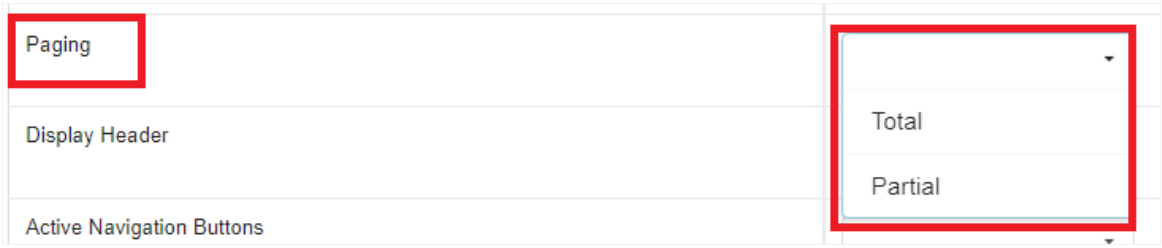


Example with display line number disabled



Pagination

This attribute defines the type of pagination used in the application: **Partial** and **total**



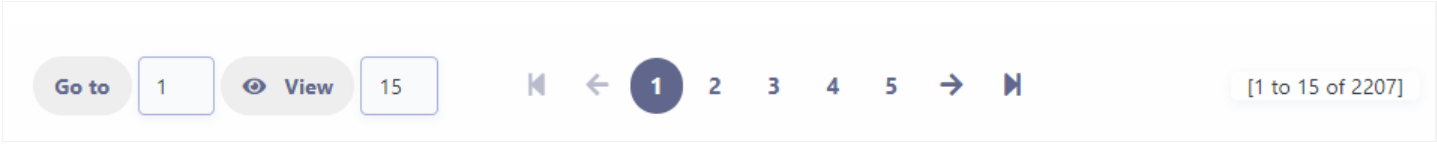
This attribute is only available for horizontal forms: **Multiple Records**, **Editable Grid** and Editable **Grid (View)**

Partial

Displays a limited amount of records per page, this amount is defined in the **Lines per Page** attribute, which is only available in this type of pagination.

Standard toolbar when using partial pagination.

When using partial pagination all navigation options are available in the application.

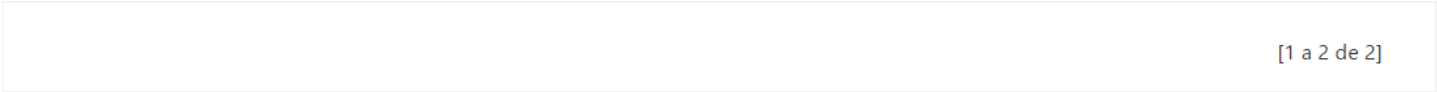


Total

Lists all listed records on a single page, removing navigation buttons.

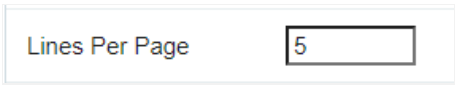
Standard toolbar when using full pagination.

When using full pagination, navigation buttons are automatically hidden from the application toolbar.



Lines Per Page

Defines the maximum total of rows that will be displayed per page.



This attribute is only available in horizontal forms (Multiple Records, Editable Grid and Editable Grid (View)) configured to use **partial pagination**.

Application example with five rows per page

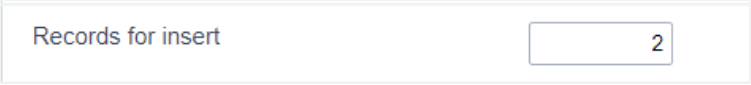
In the example below, the application is initially displayed with 5 records per page as defined by the developer.

When using the **Choose Quantity** button on the application's toolbar the developer allows the end user to define the number of lines displayed through the **View** button.

Records for Insert

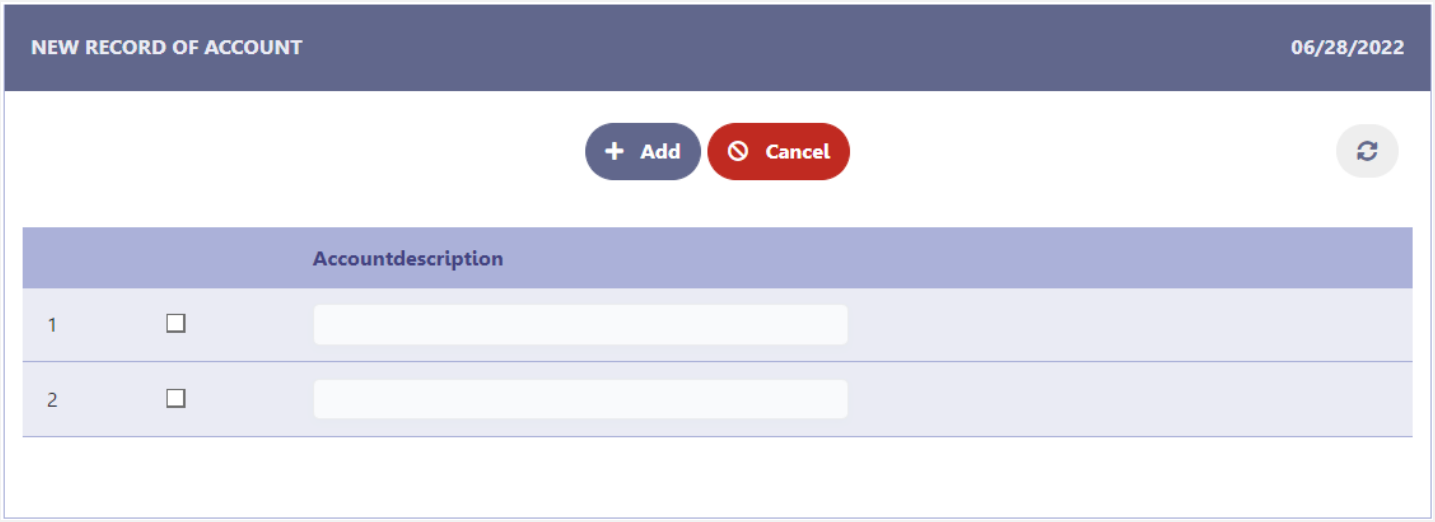
Defines the number of rows available for simultaneous inclusion of records in the multiple records form. The value defined in the **Records for Insert** attribute works regardless of the total number of lines per page of the application.

This attribute is only available for **Multiple Records** forms



Example of Multiple Records forms in insert mode

When clicking on the **New** button, the rows for inclusion of the record will be displayed according to the configuration performed in the attribute.



Line break in title

This attribute defines whether or not the line break in the field label will be allowed.

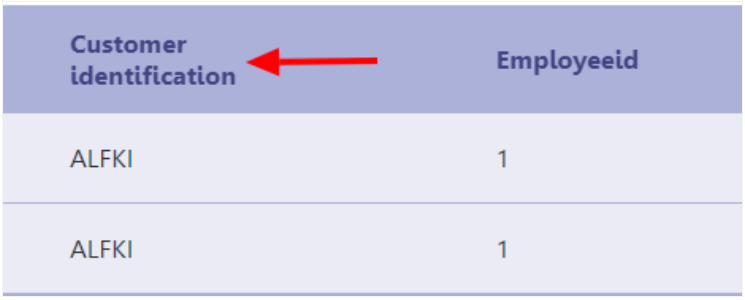
This attribute is only available in horizontal forms (Multiple Records, Editable Grid and Editable Grid (View)) configured to use **partial pagination**.

When enabled, does not allow line breaks in the field label. In this way, even if the field occupies a smaller space than the label, as in the Customer Code field in the image below, the line will not be broken.

Example of line break in title enabled

When disabled, allows a line break in the field label. When using this option, the space destined to the label will be defined by the size of the field, in this way, if the ladel is larger than the space destined to the field, there will be a break in the line.

Example of line break in title disabled



Horizontal Alignment

Defines the horizontal alignment of the application and can be defined as: **Centered**, **Left** or **Right**.

The attribute's default setting is **Centered**

Centered

Positions the application in the center of the page, respecting the margin definitions defined in the theme or in the **margin** attribute.

Application example with centered alignment

Left

Positions the application in the left corner of the screen, respecting the definitions of the margins defined in the theme or in the **margin** attribute.

Left-aligned application example

Right

Positions the application in the right corner of the screen, respecting the definitions of the margins defined in the theme or in the **margin** attribute.

Right-aligned application example

Vertical Alignment

This attribute defines the vertical alignment of the application and can be configured with the following options: No Value, Top, Center ou Bottom.

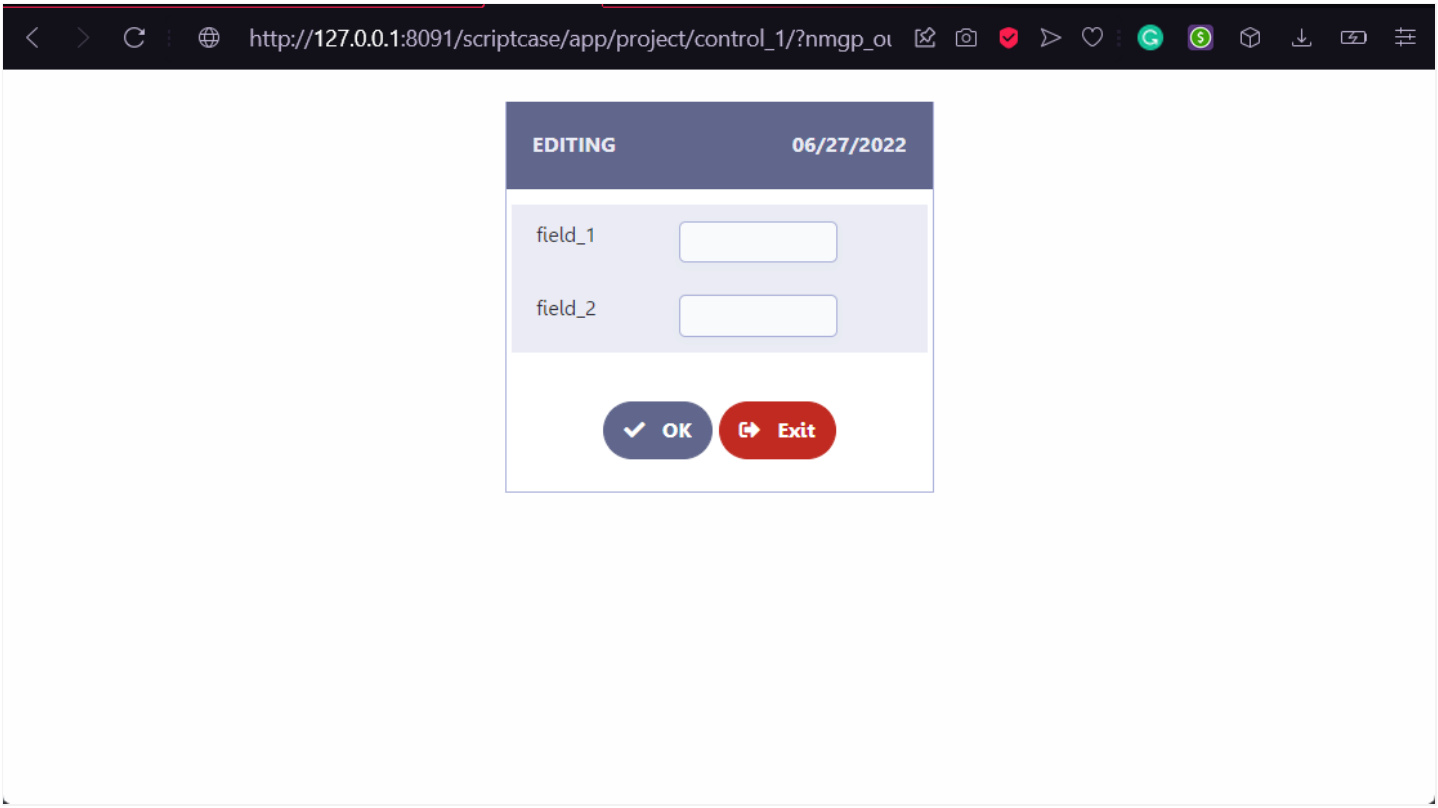
When using **Top**, **Center** or **Bottom** alignments, the margin attribute is disregarded.

See below for details on each of the alignment options.

No Value

Standard vertical alignment for new applications. In this configuration, the application is positioned at the top of the screen respecting the margins defined in the **margin** attribute or in the definition of the theme used in the application.

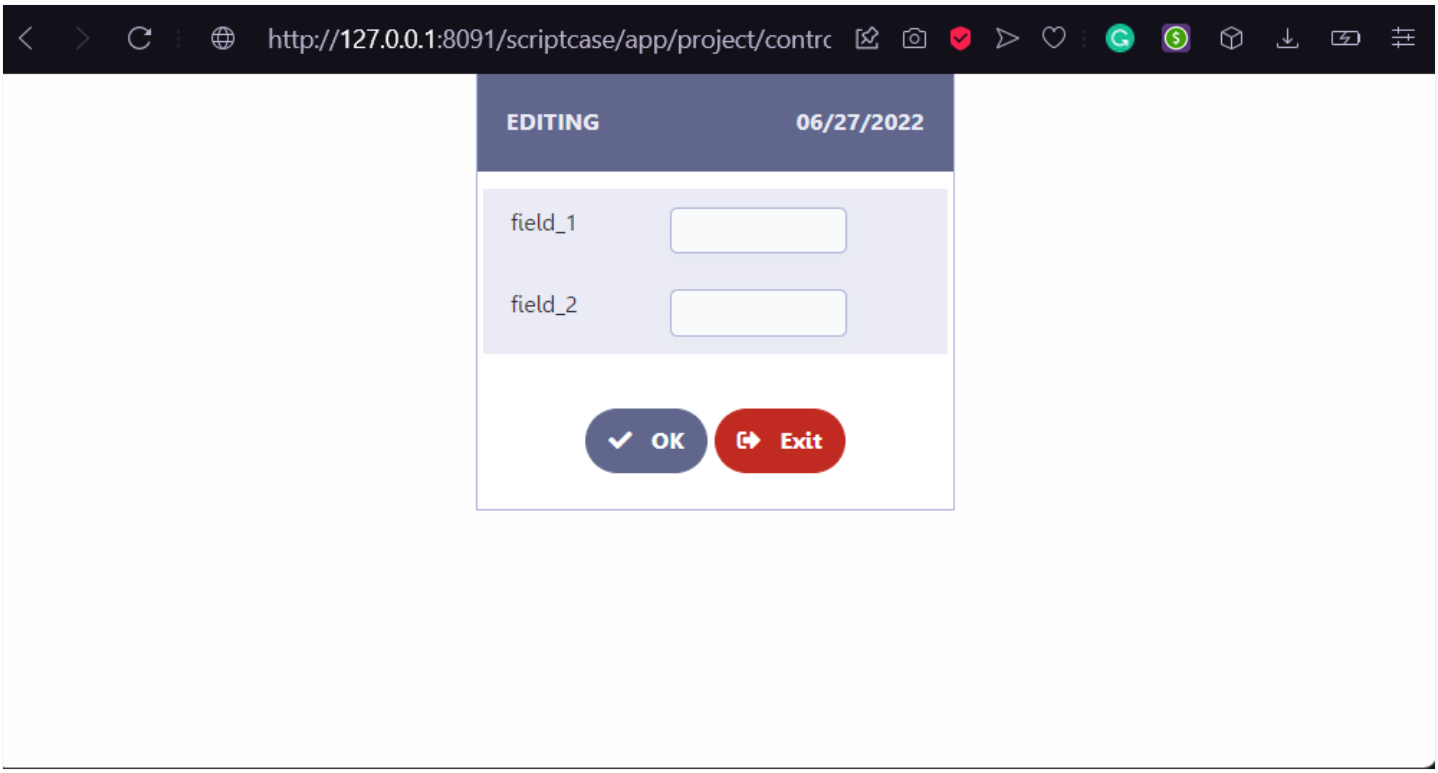
Application example with standard alignment



Top

Places the application at the top of the screen, disregarding the definitions of margins defined in the application theme or in the **margin** attribute.

Application example with alignment on the top

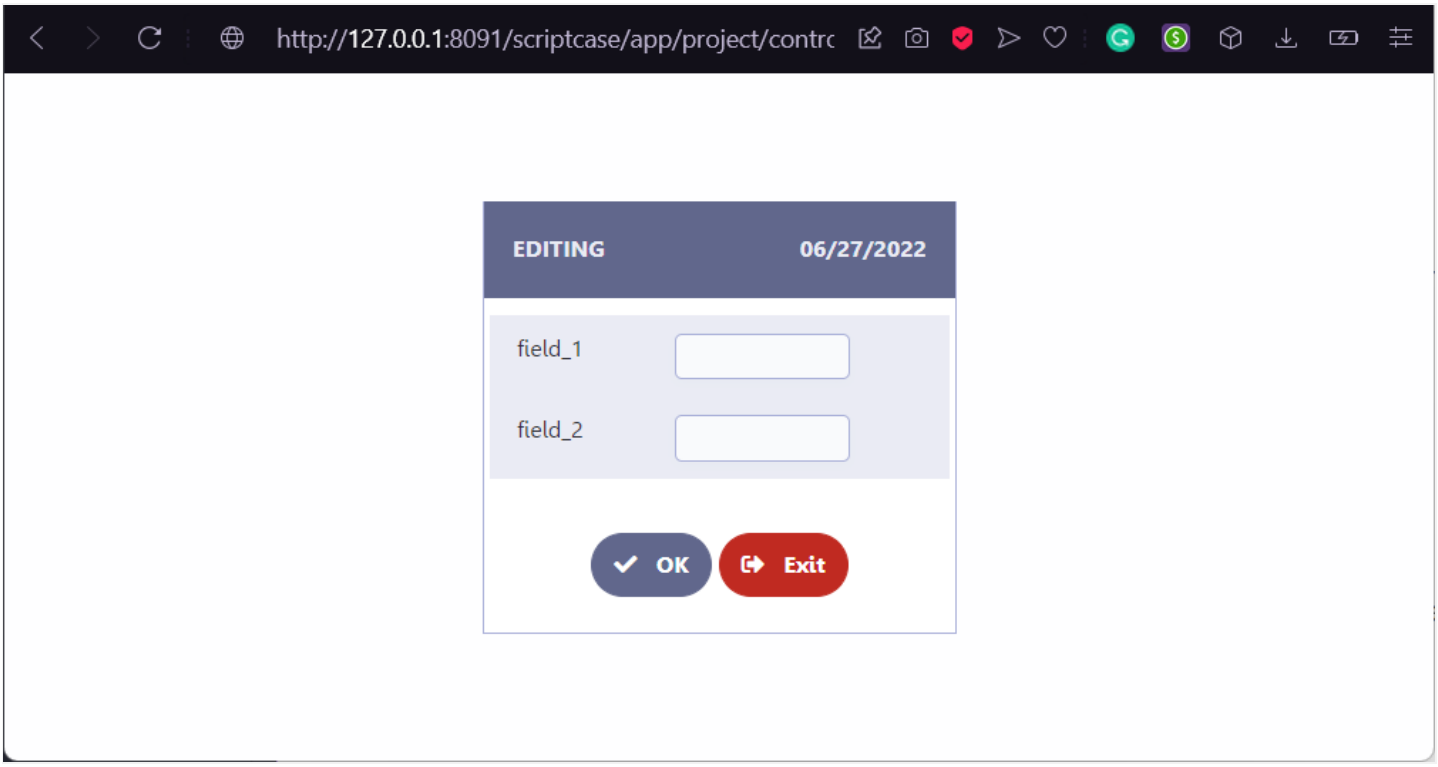


Center

Positions the application in the center of the screen, disregarding the definitions of the margins defined in the application theme

or in the **margin** attribute.

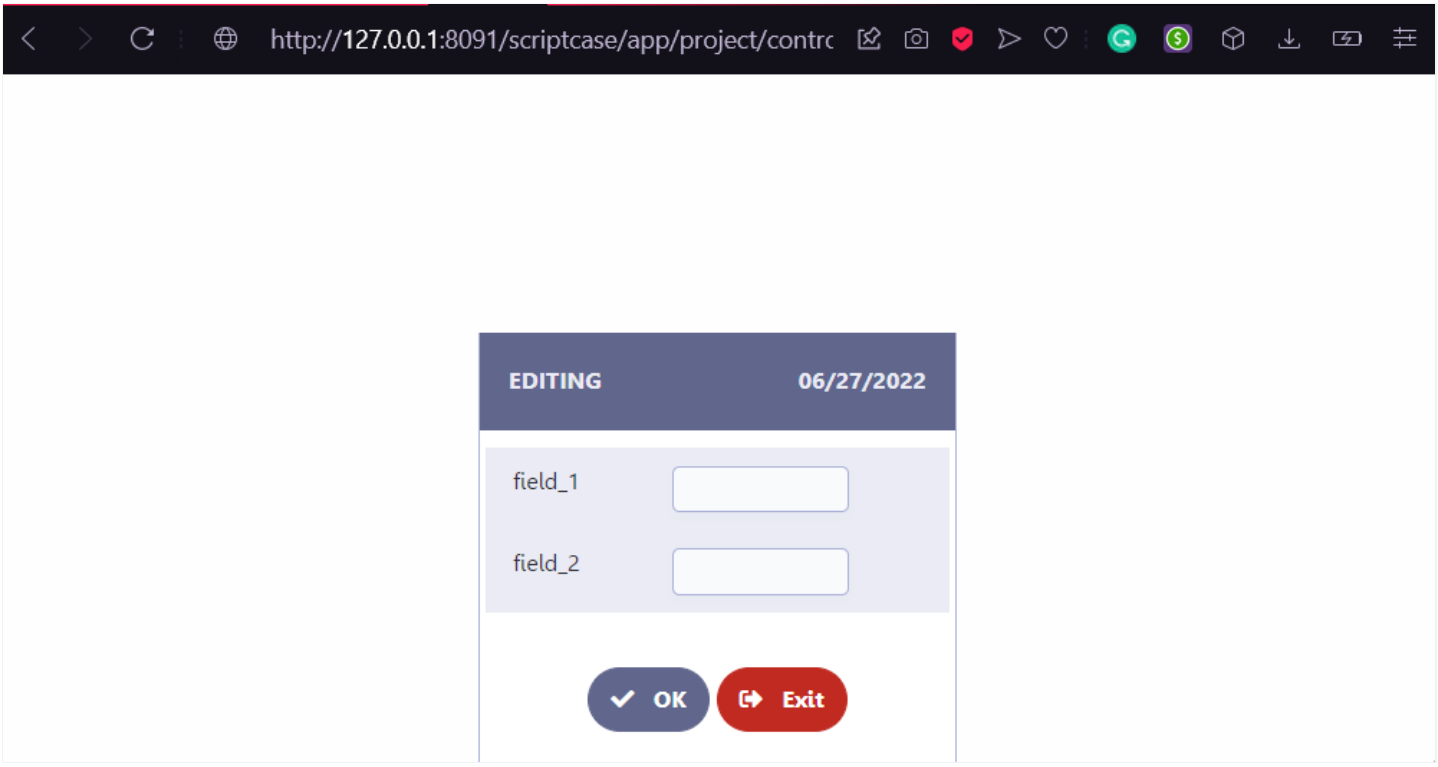
Application example with Center alignment



Bottom

Places the application at the bottom of the screen, disregarding the definitions of margins defined in the application theme or in the **margin** attribute.

Application example with alignment on the bottom



Margins

Allows the developer to set the application's margin values in pixels. If no value is specified, the application will use the default values from the configured theme.

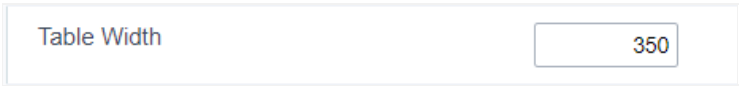
Attention:
When configuring the **Vertical Alignment** attribute, the defined margin values will be ignored.

Example of how to use the attribute

Table Width Unit

Define a unit of width used in the **Width of Table** attribute, responsible for defining the application extension.

The available options are: **Percent**, **Pixel** and **auto**.



Percentage

The application width will be defined using percentage as the width unit. In this way, when inserting a value of 50 in the **Width of Table** attribute, the width of the application will be defined with: `width: 50%;`

In this case the application will have a variable width, defined by the available space.

For example, an application defined with a width of 50%, if opened in an iframe with a width of 800px it will occupy 400px. This same application opened in a space with 200px will occupy 100px.

Pixel

The width of the application will be defined using pixel as the unit of width. In this way, when inserting a value of 50 in the **Table Width** attribute, the width of the application will be defined with: `width: 50px;`

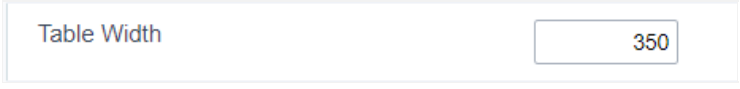
In this case, the application will have a fixed width, occupying the defined value regardless of the available space.

For example, an application defined with a width of 800px, if opened in an iframe with a width of 500 pixels, it will occupy the defined 800px, creating a horizontal scroll within the iframe.

Automatic

When using this option, the **Width of Table** attribute will be disregarded and the size of the application will be calculated by the browser.

Table width



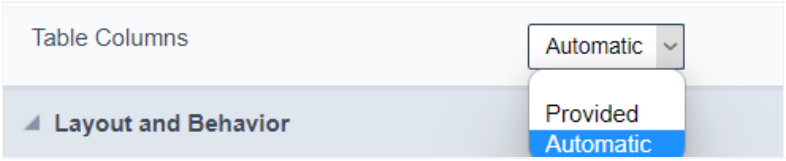
Defines the width of the application. In this attribute, only numbers must be used, the width unit that will be used is defined in

the **Unit of Table Width** attribute.

When selecting **automatic** in the Table Width Unit, the value defined in this attribute is disregarded.

Table Columns

This parameter defines the width of the application fields. The available options are: **Automatic** and **Informed**.



This attribute is only available for **single record forms** and will be ignored when using **Inputs with 100% width** option.

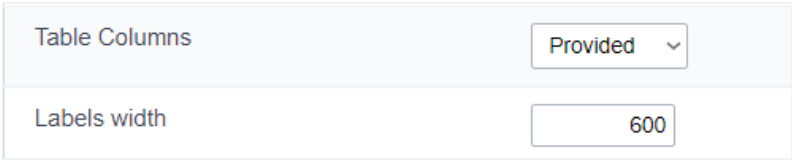
Available values in the attribute:

Automatic

The field width is automatically calculated by the browser, which will adjust according to the entered text size.

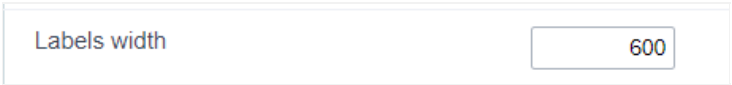
Informed

In this case, the width of the fields is defined by the value informed in the **Width of labels** attribute, which uses pixel as the width unit.



When selecting the informed value, the Width attribute of the labels is mandatory, if no value is informed, the generator will disregard the **Table Columns** attribute.

Labels Width

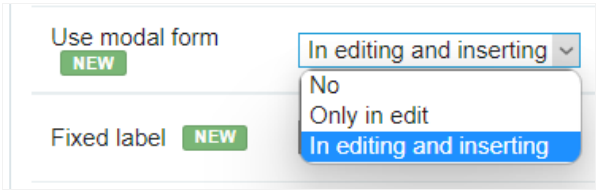


Defines the width of the application's labels, using pixel as the width unit. To set the width, just enter the numbers as in the image above.

This attribute is only available when the **Table Columns** attribute has the value **informed**.

In this case, the **Labels Width** parameter becomes mandatory, if no value is informed, the **Table Columns** attribute will be disregarded.

Use modal form



Only available in editable **Grid (View)** forms, this attribute defines the application's behavior for adding and updating records.

The available options are:

- No - Inserting and updating of the record online.
- Only in edit - Inline inserting and record update in modal form
- In editing and inserting - Adding and updating the record in the modal form

For the **Only in edit** and **In editing and inserting** options, two new attributes will be displayed for defining the width and height of the modal where the form will be opened.

See each of the detailed options below.

No

Default value of the attribute when creating a form, in this configuration the updating or insertion of data will be carried out in line.

Example of in-line registry editing

By clicking the edit icon the fields will be enabled according to your configuration.

	Orderid *	Customer identification	Employeeid	Orderdate mm/dd/yyyy	Freight	Shipname	Priceorder
	885	ALFKI	1	02/15/2022	0.00		0.00

Example of adding an online record

By clicking the **+ New** button a blank line will be added after the last record listed.

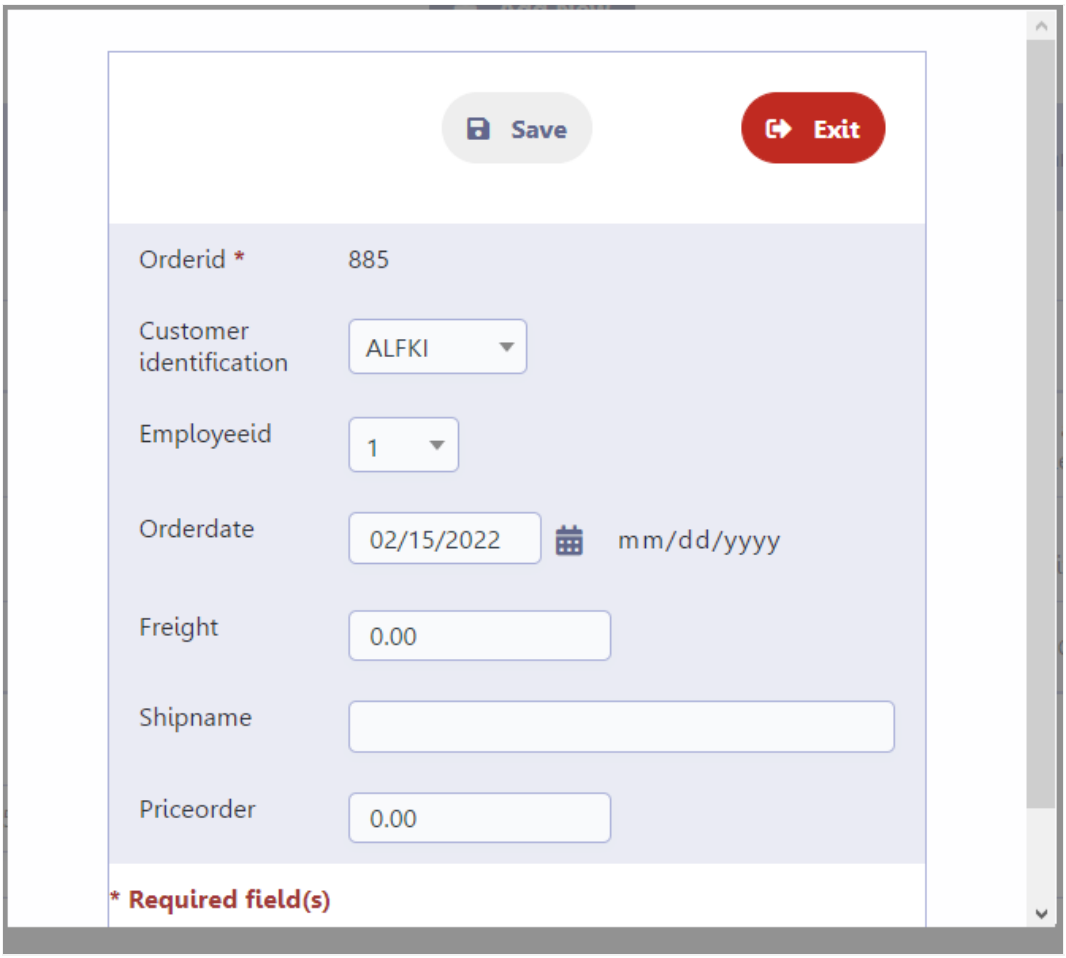
	Orderid *	Customer identification	Employeeid	Orderdate mm/dd/yyyy	Freight	Shipname	Priceorder
	885	ALFKI	1	02/15/2022	0.00		0.00
	886	ALFKI	1		0.00		0.00
	10,248	SANTG	4	01/30/2019	15.73	Vins et alcools Chevalier	10,574.00
	10,249	ANATR	7	01/30/2019	14.52	Toms Spezialitten	229.90
	10,250	HANAR	4	01/31/2019	79.86	Hanari Carnes	96.00
		ALFKI	1				

Only in edit

When selecting the option **Only in edit**, when clicking the edit icon a modal will open with a form for editing the selected record. In this configuration the record inclusion remains in line.

The size of the modal is defined in the parameters **Modal Width** and **Modal Height**. Both use pixel as a unit of width, just informing numbers.

Record editing example in modal



In editing and inserting

This option defines the use of the modal form for inclusion and editing of records.

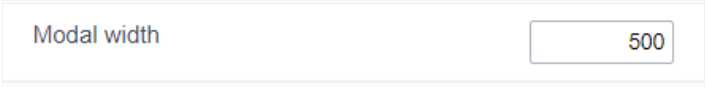
The size of the modal is defined in the parameters **Modal Width** and **Modal Height**. Both use pixel as a unit of width, just informing numbers.

Using this setting in the **Use modal form** attribute allows the developer to individually define the fields that will be displayed in the modal form and those displayed when accessing the application. In practice, it is as if there were two applications, one for editing and including records open in the modal and another for displaying and deleting records.

Field configuration must be performed in the Field Positioning menu, check our documentation.

For more details watch the video Form horizontal (edição e inclusão com modal)

Modal Width



Set the width of the modal where the form will open. The attribute uses **pixel** as a unit of measure, so only numbers must be informed to define the width.

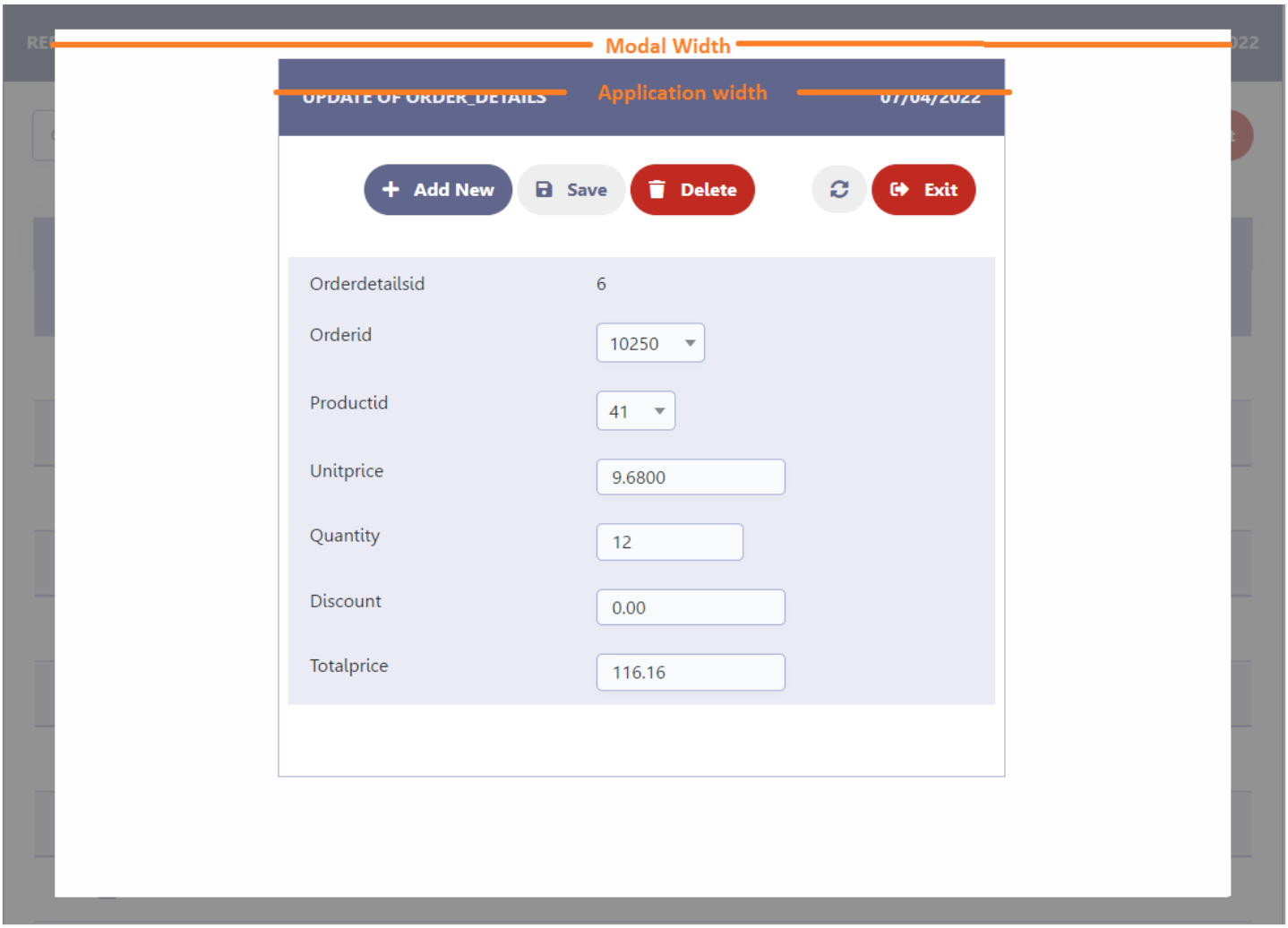
This attribute is only available for **editable grid forms(view)** where the attribute **Use modal form** is set to: **On Editing** or **On**

Editing and Inserting.

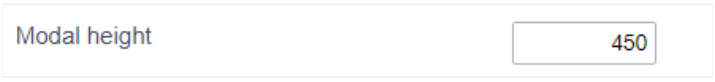
The width of the application opened in the modal respects the definition of the **Width of Table** attribute.

Example of the application opened in the modal

This attribute controls the **Modal Width**, which is basically the space available for the form to be displayed. The width of the application that will be opened respects the configuration of the **Width of the Table** of the Editable Grid Form (view).



Modal Height

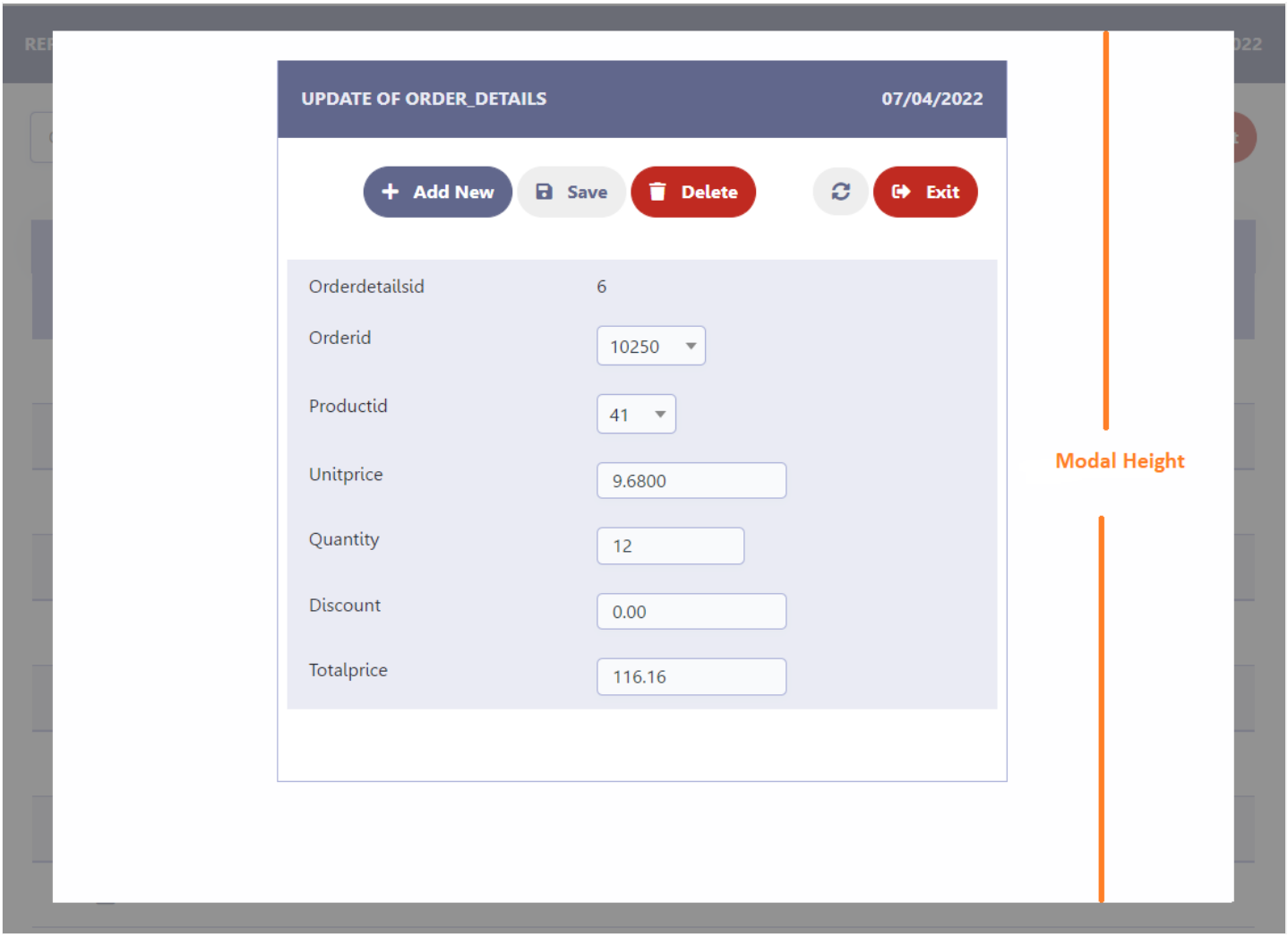


Defines the height of the modal where the form will be opened. The attribute uses **pixel** as a unit of measure, so only numbers must be informed to define the height.

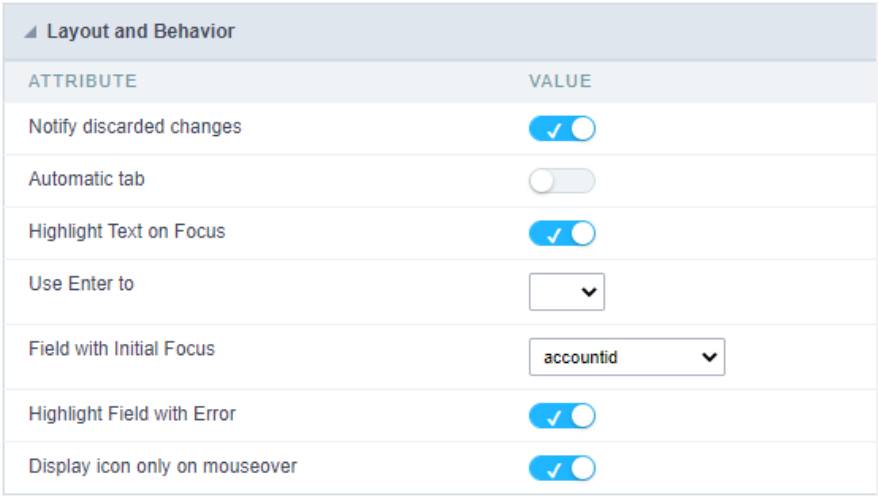
This attribute is only available for **editable grid forms(view)** where the attribute **Use modal form** is set to: **On Editing** or **On Editing and Inserting**.

Example of the application open in the modal

This attribute controls the **Modal Height**, which is basically the space available for the form to be displayed. The height of the application that will be opened respects the number of fields selected for the modal form.



Layout and Behavior



Check the details of each attribute available in the Layout and Behavior block below.

Notify discarded changes

This attribute defines the behavior of the application regarding changes that have not yet been saved if the user performs some **reload of data** action.

Data reload examples

- Use reload button .
- Navigate between pages.
- Redirects to another application in the same window.

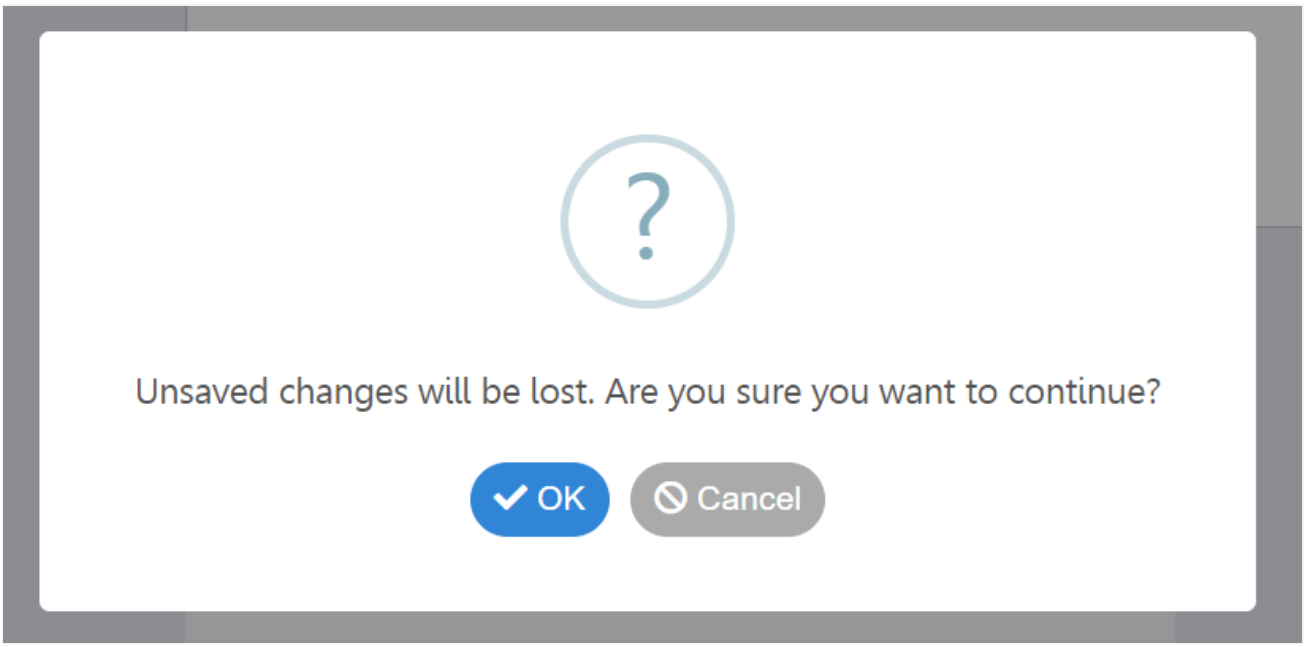
When Disabled, if the user performs any action that causes the data to be reloaded before the changes made are saved, the initiated action will be executed and the unsaved data will be lost.

When enabled, whenever the user performs any action that causes the data to be recharged before it has been saved, an alert will be displayed informing about the possible loss of this data.

In this case, the user can choose to click **OK** which will continue the action started, culminating in the discard of unsaved changes or click **Cancel** which will cause the action started to be interrupted, keeping the unsaved data.

Example of discarded change notification

Some actions may cause the loss of unsaved data, for example: Navigating between records or using the reload button

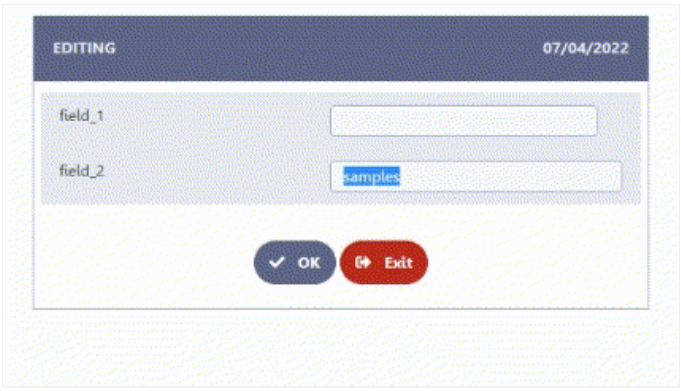


Automatic tab

Shifts focus to the next field when the number of characters in the current field reaches the limit defined in the settings.

Example of auto tab

In the example below, **Field 1** has a limit of 3 characters. When filling in all the characters the focus changes to **Field 2**.

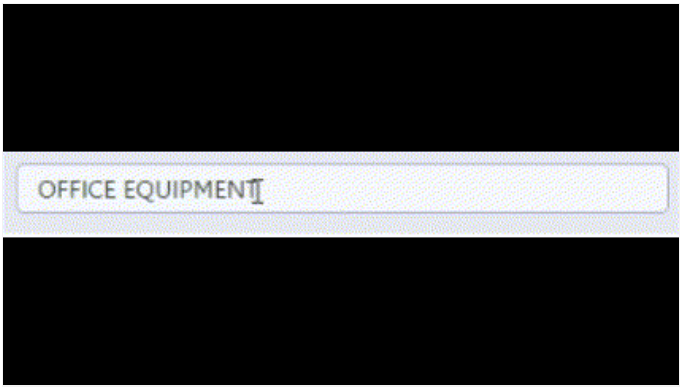


Select Field Content

Defines whether the field content will be selected or not when receiving focus.

When enabled all field contents are selected..

Example of field behavior with attribute enabled



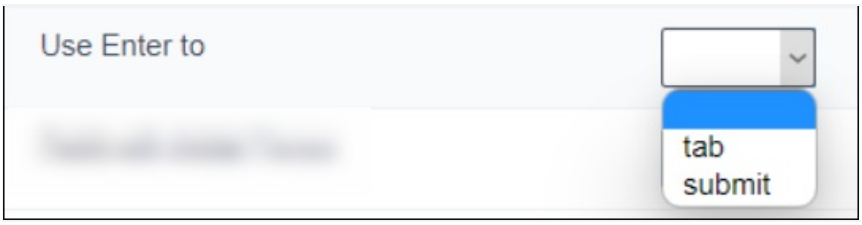
When disabled the field content will not be selected.

Example of field behavior with attribute disabled

Use the Enter key to

This attribute defines the behavior of the enter key in the application, having as options: **tabular** and **submit**.

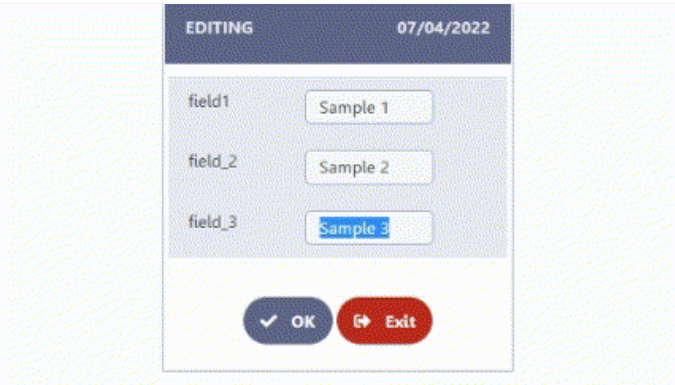
The **submit** property is not available for form applications.



Tab

Allows the user to navigate between application fields using the enter key. The tabs follow the field order defined in the **select fields** menu.

Tab example with enter



To submit

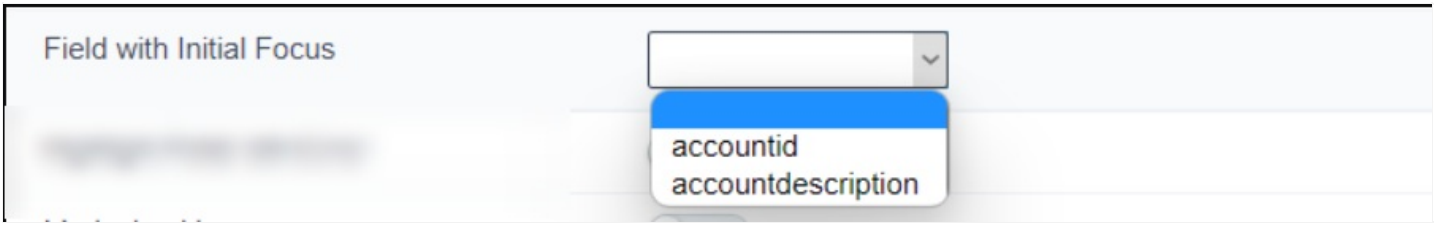
Allows the user to submit the application using the enter key.

Example submit with enter

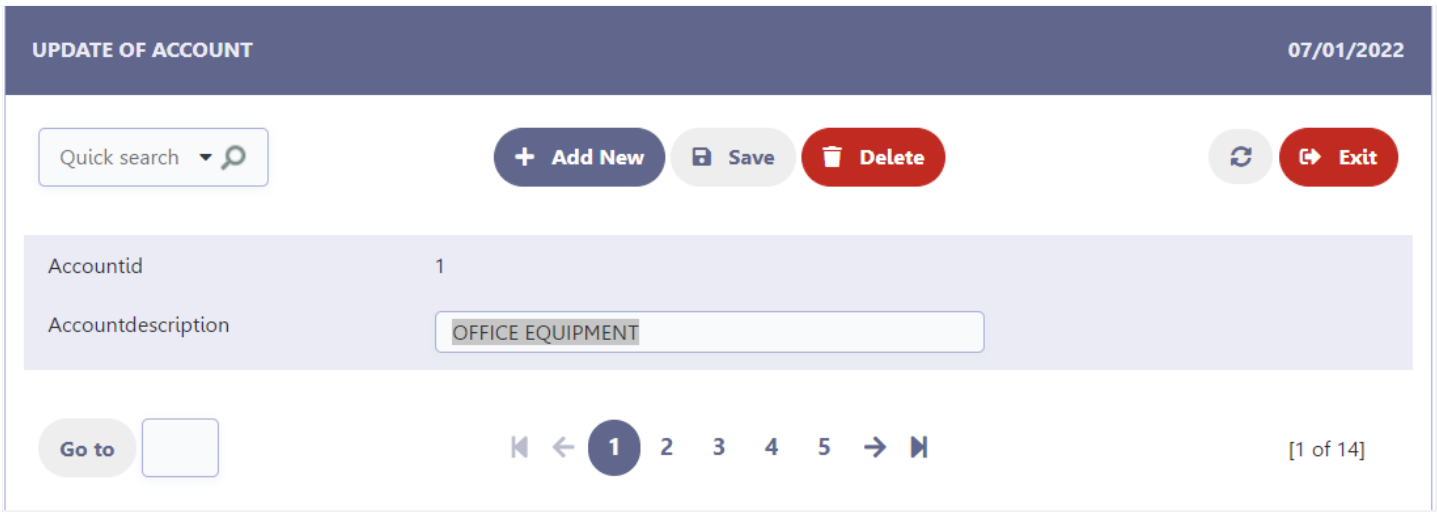
This option is not available for form applications.

Field with initial focus

Defines the field that will receive focus when the application is loaded. Enabling the user to start typing without having to use the mouse to focus on the field.



Example of initial focus



Focus field with error

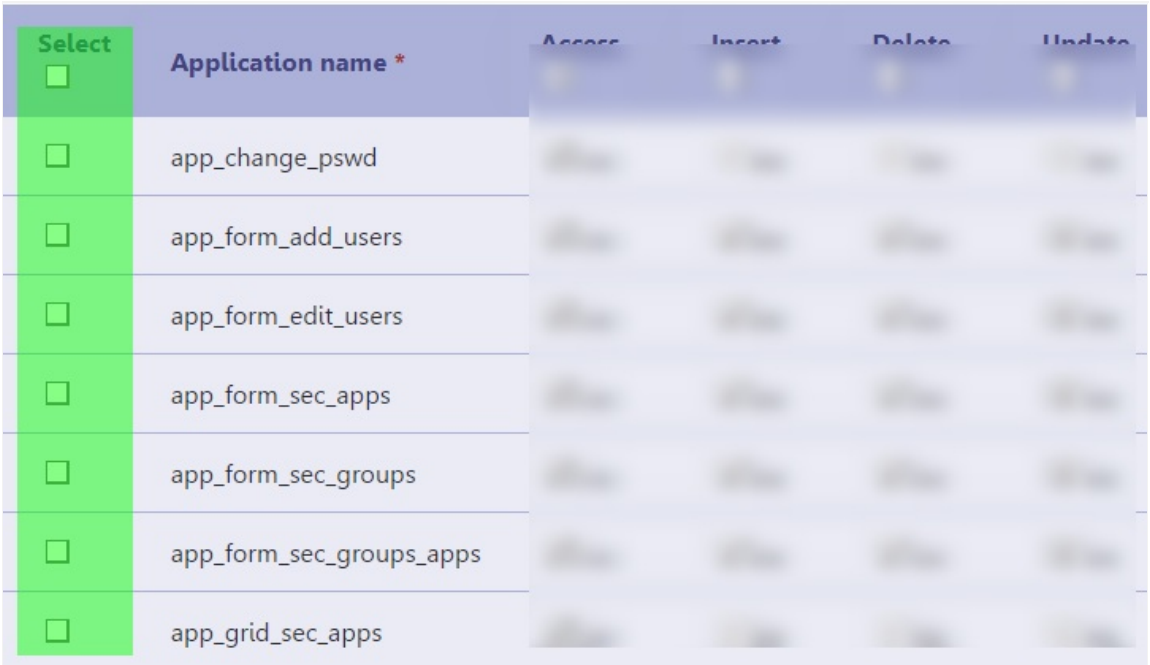
Defines the behavior of the focus in relation to the fields that present an error when submitting the form.

When enabled, when submitting the form, the field with an error will receive focus. If more than one field has an error, the focus will be on the first field with an error, following the order defined in the **Fields Position** menu.

When disabled, when submitting the form, no field will receive autofocus in case of an error.

Check all checkboxes

This attribute controls the display of the **Select** column, allowing application users to check all the checkbox fields of the application or just a specific record.

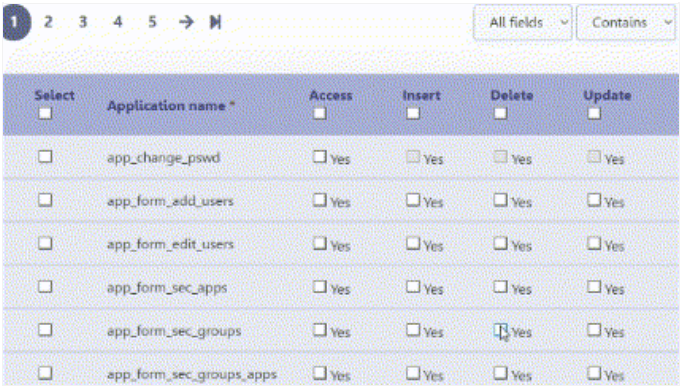


This attribute is only available in **Multiple Record** forms

This column allows two different functions depending on the checked box, check the examples below.

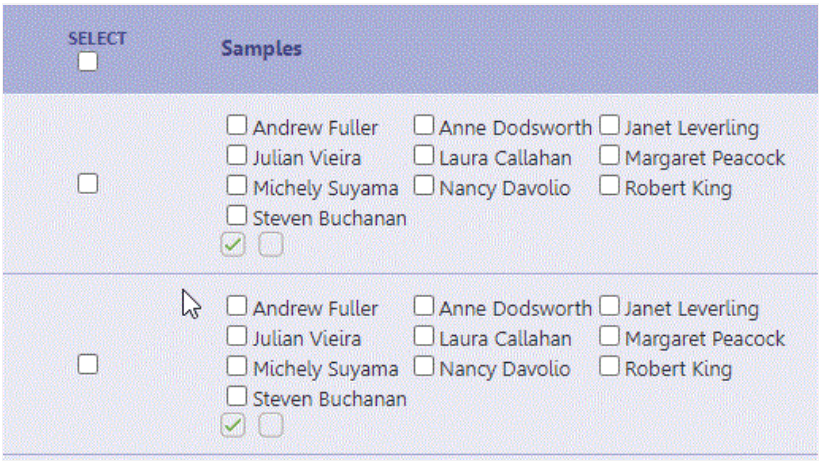
By checking the checkbox below the Select label

In this case, all checkbox fields of the current application page will be checked.



When checking the checkbox in the record line

In this case, all fields of the record configured as checkboxes will be checked.



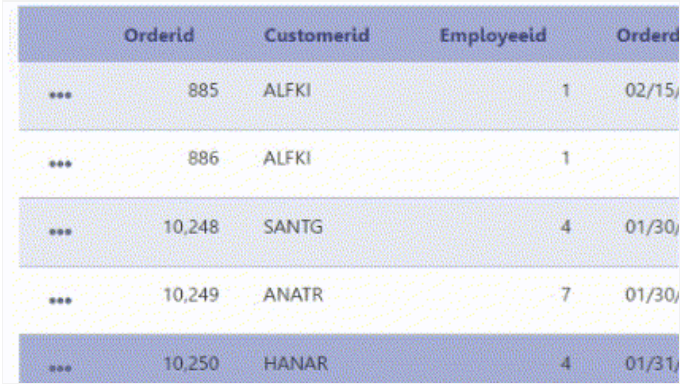
Display icon only on mouseover

Sets whether the theme sort icons will always be displayed or whether they will only be displayed on mouseover.

This attribute is only valid in Horizontal forms: **Multiple Records**, **Grid Editable** and **Grid Editable(View)**.

When Enabled, sort icons are displayed on mouseover only.

Sort icons displayed on mouseover



When Disabled, the sort icons will always be displayed in the application.

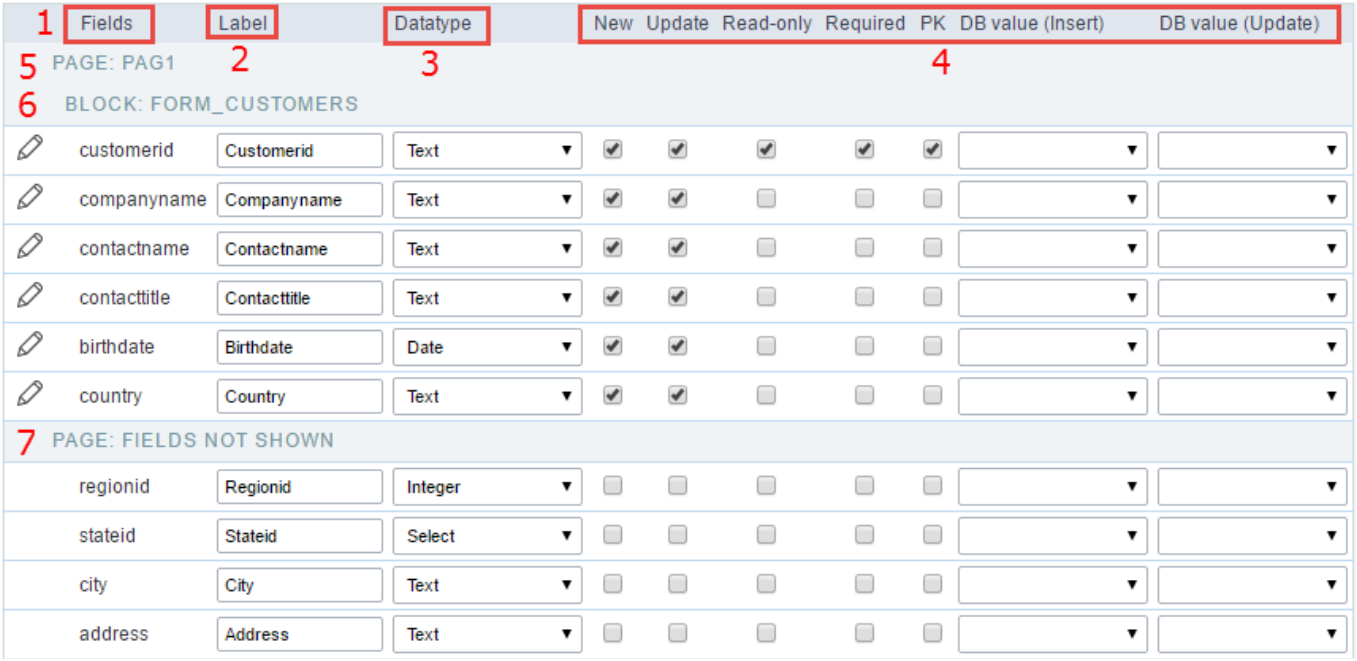
Sort icons always displayed

Orderid	Customerid	Employeeid	Orderdate
885	ALFKI	1	02/15/2022
886	ALFKI	1	
10,248	SANTG	4	01/30/2019
10,249	ANATR	7	01/30/2019
10,250	HANAR	4	01/31/2019

Form Edit fields

This interface is useful for editing the field settings and their position to display.

Edit Fields



Edit Fields Configuration.

Fields

It allows accessing the field settings (a pencil icon on the left). You can change the field position by dragging them to the desired position. Drag a field to “fields not displayed” if you don’t want it in the app.

Label

It defines the title of the field in the app. For example: if the field name in the database is fld_txt_customer_name, you can display the label “Customer Name”.

Data type

It informs the data type of the field.

New

It defines if the field is available when inserting new records.

Update

It defines if the field is available when updating records.

Read-Only

It defines the field as a label. The user can’t change its value.

Required

It defines if the field must contain a value.

PK

It defines the Primary Keys fields.

DB value (Insert)

Defines a default value for the field when inserting a new record, like an auto-increment, Date, DateTime, or IP.

DB value (Update)

Defines a default value for the field when updating a record, like an auto-increment, Date, DateTime, or IP.

Page

It shows the pages available in the application. All apps have a page, at least. Each page contains one or more blocks.

Blocks

It shows the blocks available in the application. Blocks contain fields. All apps have a block, at least. A block is displayed if it contains one or more fields.

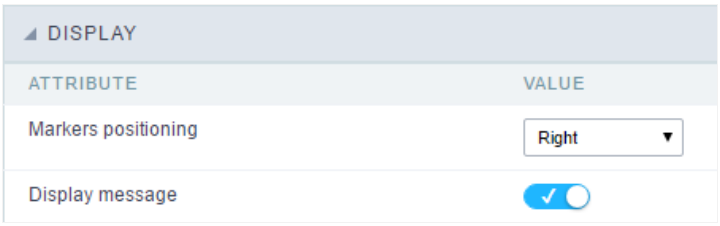
Page Fields Not Shown

Here we can see the fields that are not in the application.

Observe that you can drag any line to the desired position, blocks, and page. Pages contain Blocks, and Blocks contain fields.

Display Settings

Settings of messages display.



Display Interface.

Attributes #### Markers positioning Set the position of the markers that Indicates required fields.

Display message
Set it if you want to display the message of the required field.

> **Related Videos** ▷

Form Fields Positioning

It allows the definition of the fields that will be part of the form and its display order.

When using the Editable Grid(View) format, it is also possible to define the fields for visualization and data manipulation independently.

Form Fields

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

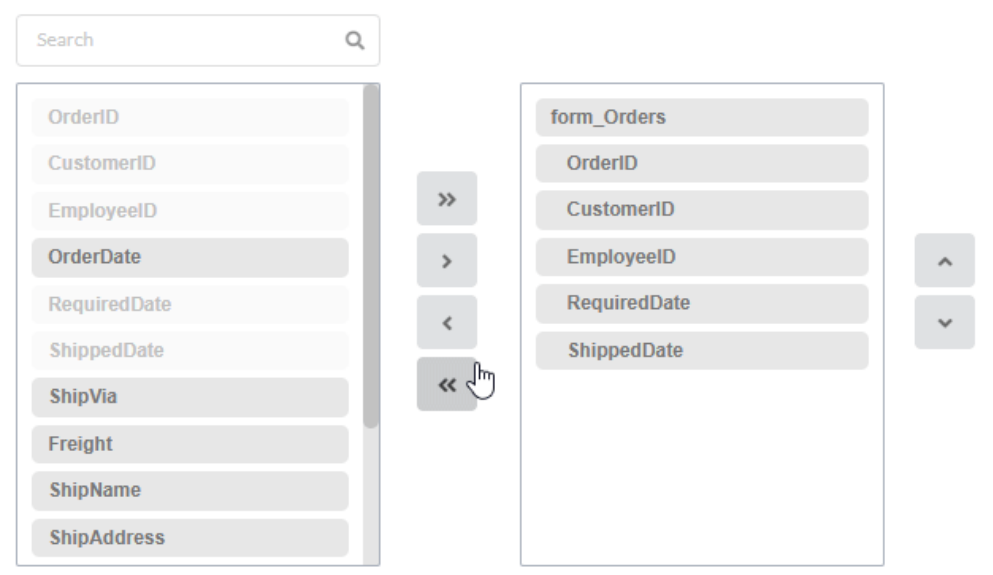
- » - Move all fields to the right.
- › - Moves only selected fields to the right
- ‹ - Moves only the selected fields to the left.
- « - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the › button.

To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning



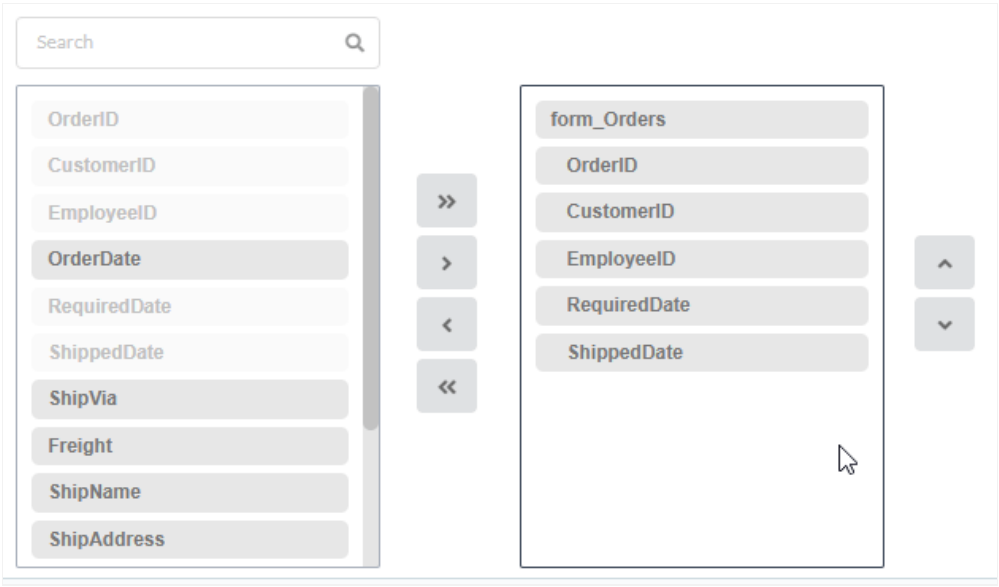
Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons ^ and ⌄ which will move the selected fields in the column.

Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

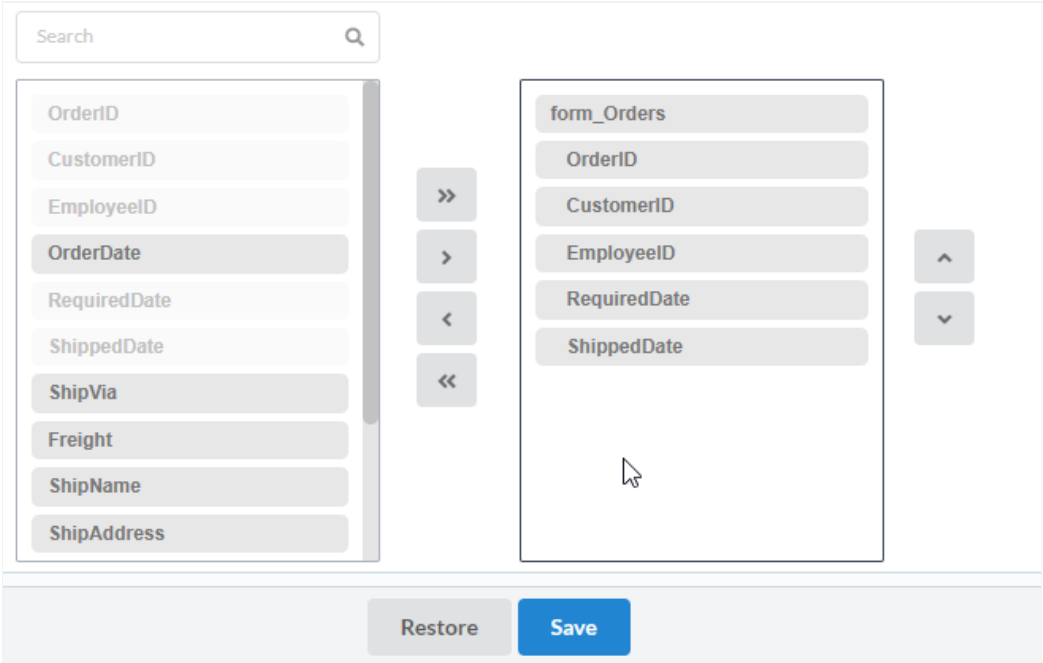


Save and Restore

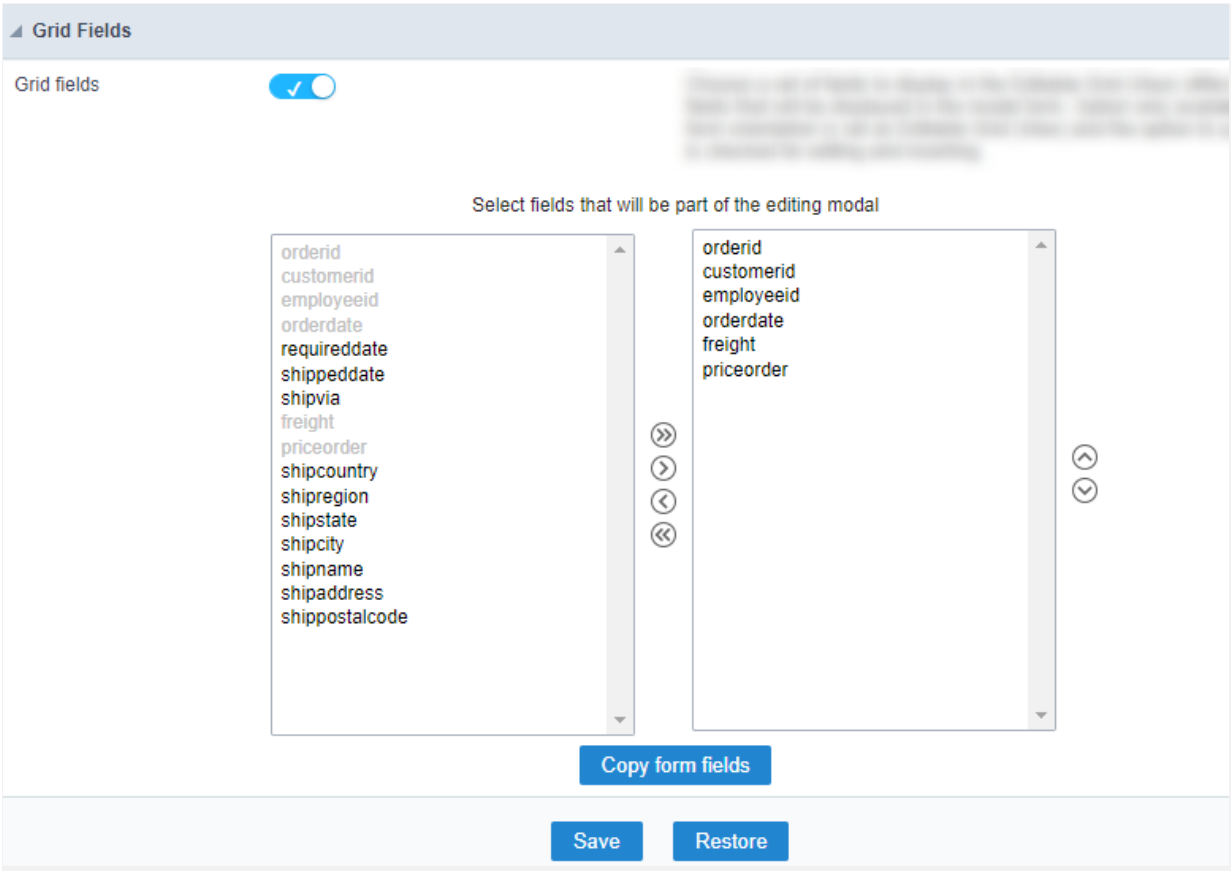
Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

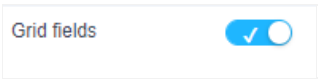
Restoring the positioning of the fields to the last saved definition.



Grid Fields



Grid fields



Allow the independent setting of the displayed fields in the Editable Grid (view) and the available fields to edit in the modal.

This option is available if the **form orientation** is set as *Editable Grid (view)* and the **Use modal form** attribute is set as *In editing and inserting*

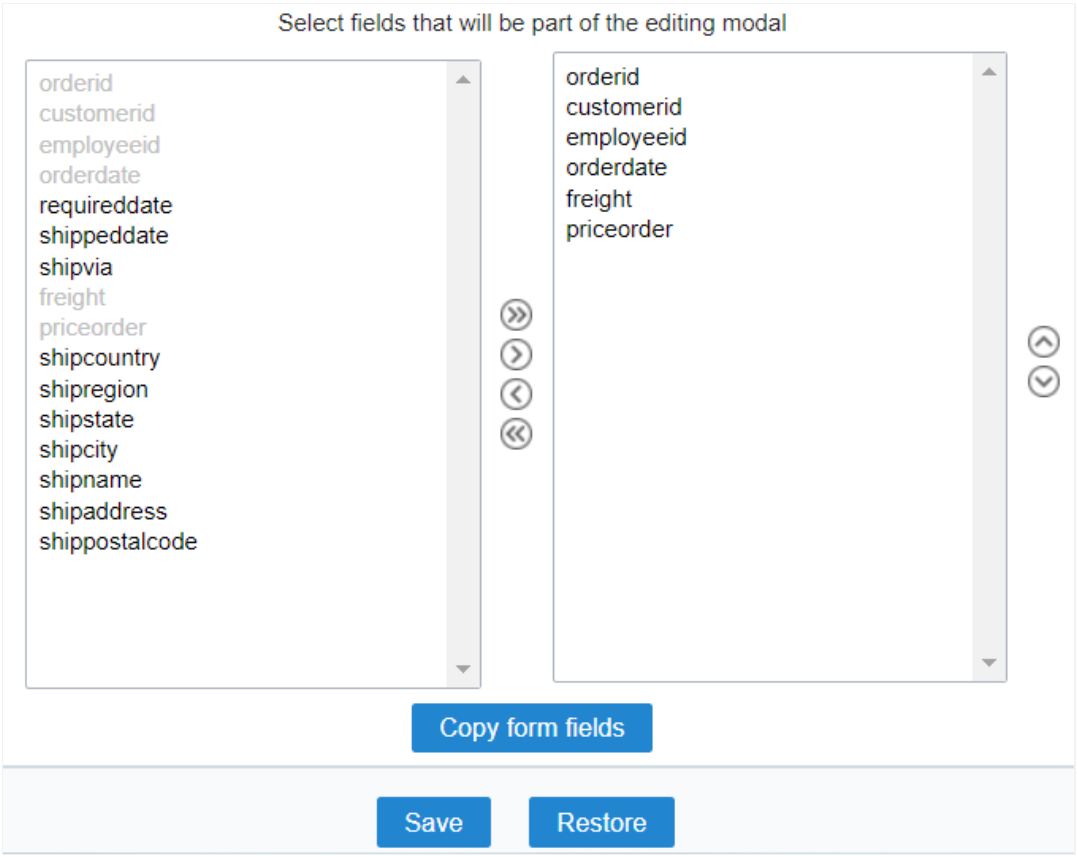
When enable

The option **Select fields that will be part** of the editing modal is displayed, to you can set the fields that showed only in the Editable Grid (view)

When disable

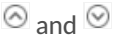
In this case, the fields display settings respect the definition made in the block **Form Fields**.

Select the fields to displayed

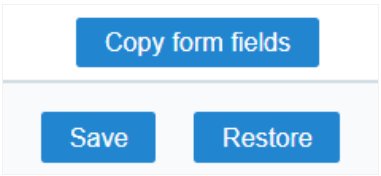


Using the positioning buttons, , it's possible set the fields that will be used only for displaying data in the application.

The fields set only to displaying will be shown in the Fields menu, thus allowing these fields to be configured.

You can also define the display order of the fields using the positioning buttons:  and

Buttons



- **Copy form fields** - Allows you to copy the fields defined in the **Form Fields** option
- **Save** - Saves the current field configuration.
- **Restore** - Restores the settings to the last save performed.

Related Link

Related Videos

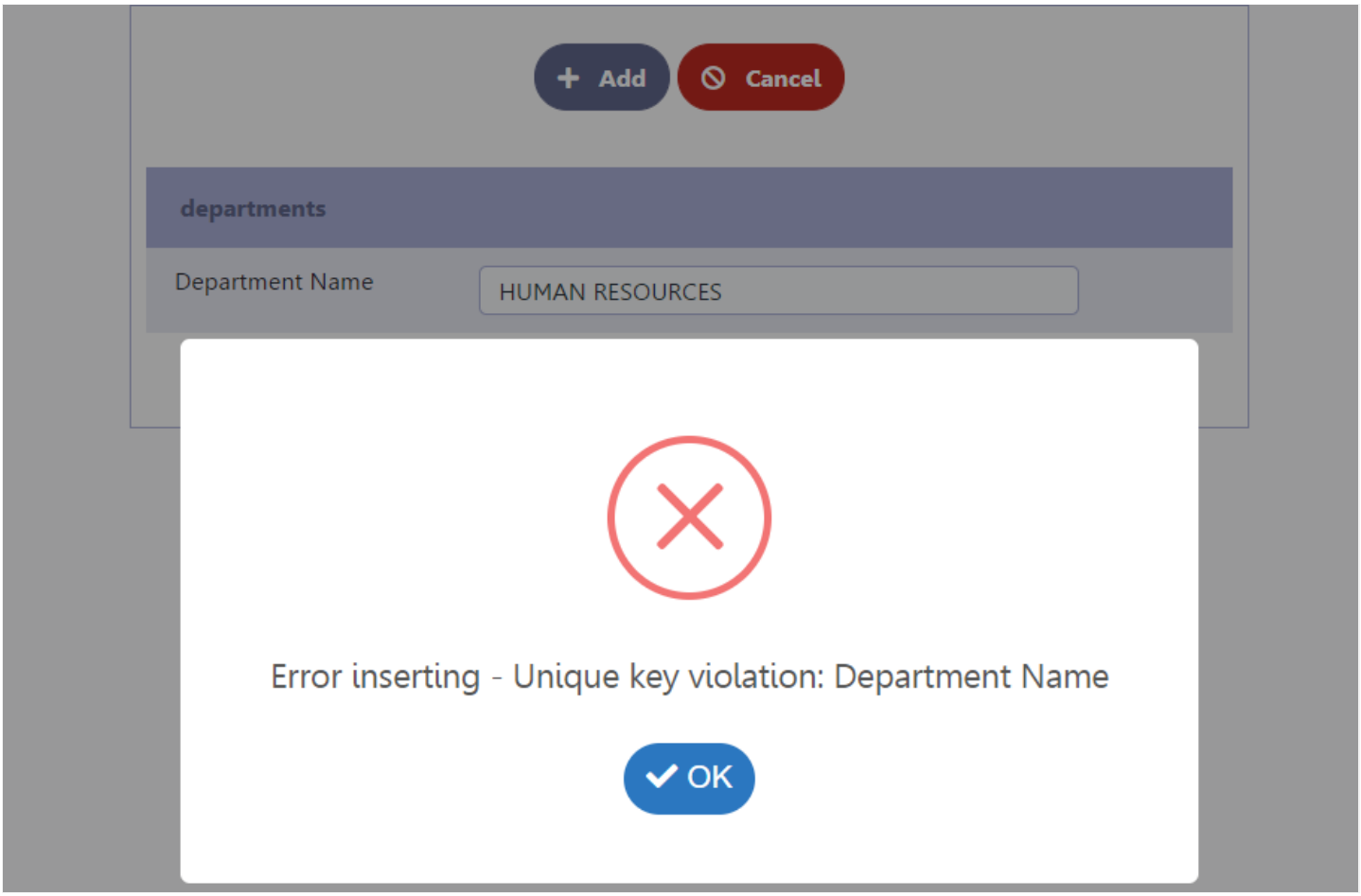
Unique key

With the unique key option you can inform which fields' data should not have repetitions within the database.

Unique key configuration Interface

If you want to combine two or more fields as unique, select the fields by holding the CTRL key of your keyboard and then click on the "Add" button.

The user will receive message if tries to insert a repeated value into the application. The validation and message are created automatically by Scriptcase, you are able to customize it by accessing the application language option and searching for the lang variable: **{lang_errm_inst_uniq}**. This lang variable contains a pre-set error message content.



Unique key message example

Related Link

Related Link

Related Video ▷

Form Toolbar

The application toolbar has two segments: Top and Bottom, in a way that is possible to define to display buttons into both areas. Those areas work independently, allowing them to display the same button, for example.

It's also possible to select the buttons and their position if the application is running on a mobile device.

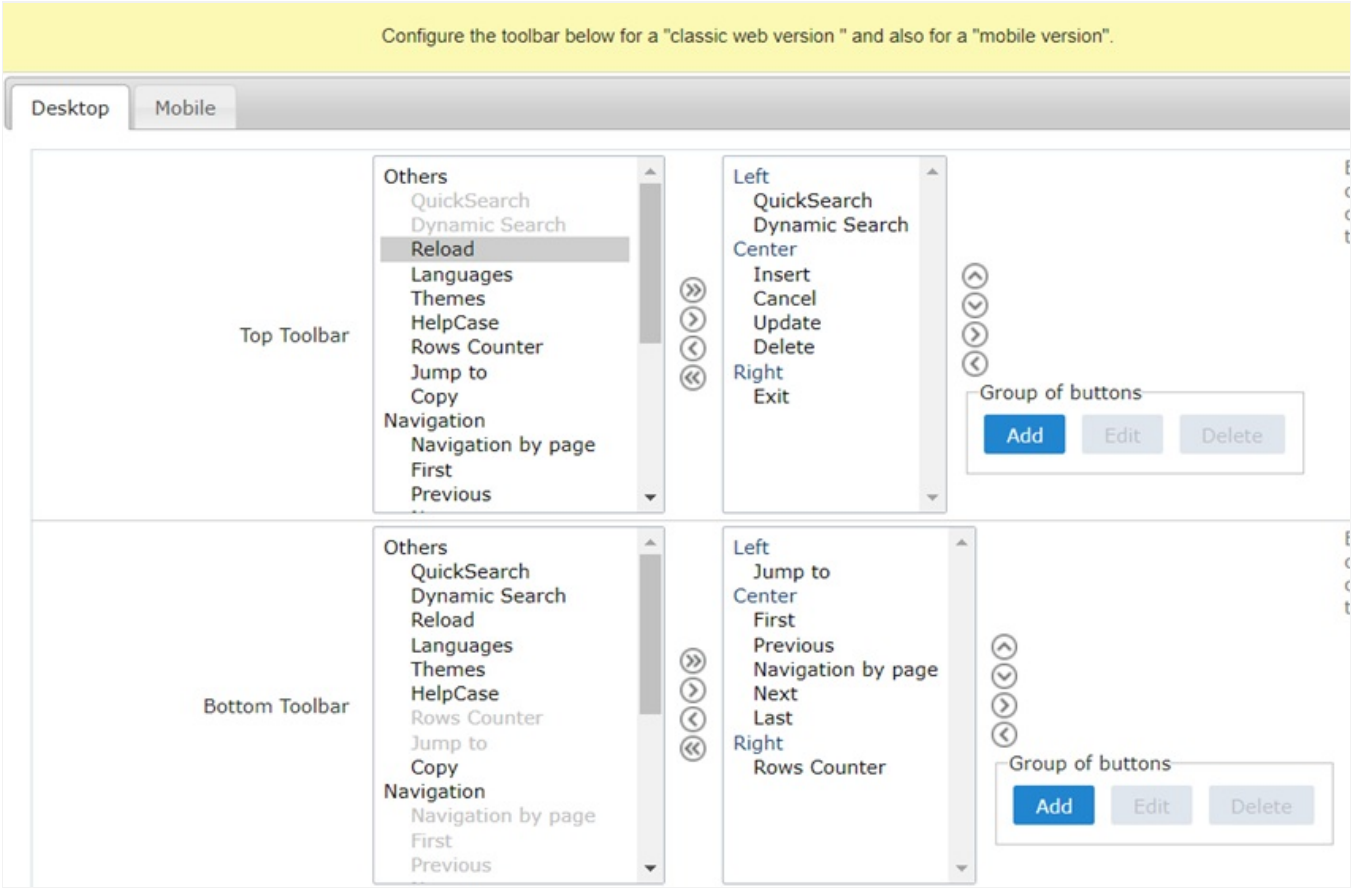
Toolbar

Desktop

Here we must inform the toolbar settings for the "Classic Web Version" mode and which buttons are available in the application when accessed from a **Desktop** environment.

Mobile

Here we must inform the toolbar settings for the "Mobile Version" mode. That is which buttons are available in the application when accessed from a **Mobile** dispositive.



Navigation:

Buttons relative to the navigation of the application.

Next	Move to the next page that can be a single record or a list of records.
Previous	Returns to displays the previous page records or a single record.
First	Move to the First page or record
Last	Move to the Last page or record
Exit	Close the application
Navigation by page	Displays a “page-number” navigation bar. **Example:** 1 2 3 4 5
Reload	Displays a button to reload the query data

Export:

The options available to export the Records. Scriptcase generates the following export formats for Forms:

PDF	Generates all the data of the application in a PDF format.
Print	Creates an HTML with the records ready for printing.

Update:

The CRUD options available in the Form.

Insert	Inserts the record into the database.
Update	Saves the changes made in a record.
Delete	Deletes the selected record.
Cancel	Cancel the changes made in a record before the insertion.

Others:

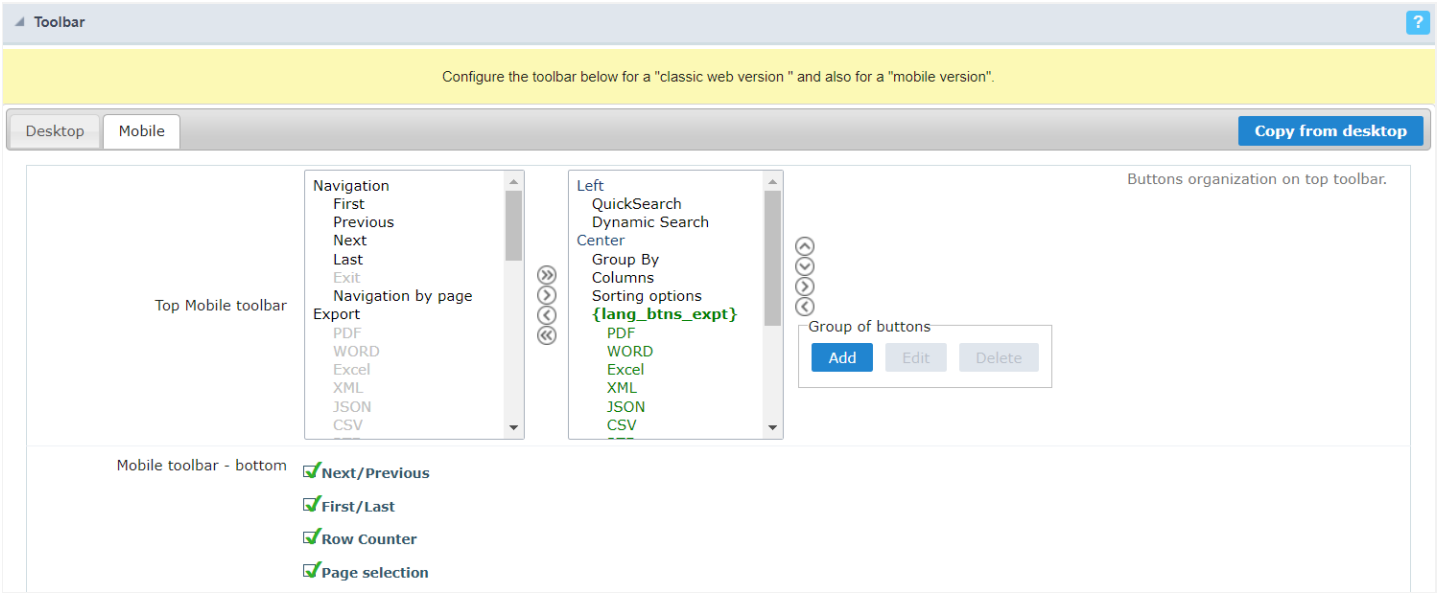
Other options available in the Form application.

Jump To	Move to the informed page or record.
Copy	Copy the current record data to another one.
Quick Search	Perform a quick search in the records of the application.
Dynamic Search	It displays the fields of the search to filter the records.
Languages	Displays a Combobox with the languages available in the project properties.
Themes	Displays a Combobox with the languages available in the project properties.
Rows Counter	Displays the number of records retrieved in the application.
HelpCase	Displays a button to open the help page.

Separator:

————————	Displays a line separating the buttons.

Toolbar Mobile



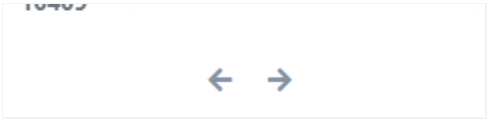
Top Mobile toolbar

It has the same options as the **Desktop** version, adding only the item “Copy from desktop”, which, when clicked, makes a copy of the items from the upper toolbar of **Desktop** to **Mobile**.

Mobile toolbar - bottom

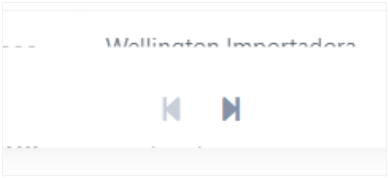
Next/Previous

Enables navigation to the next and previous page on mobile devices.



First/Last

Enables first and last page navigation on mobile devices.



Row Counter

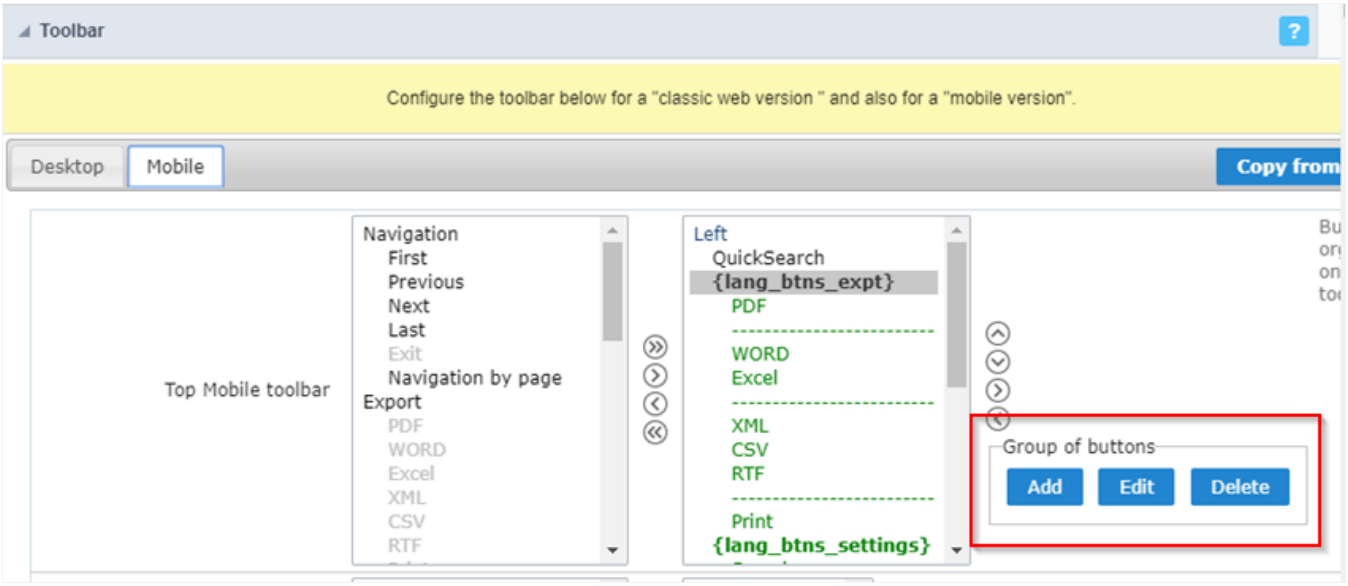
Enables the record counter showing the application's total records

Page selection

Enables page navigation on mobile.

Bottons Group

The **Group** option allows you to group a set of buttons of the application toolbar to display them as a dropdown, for example.



Add

Add a new group of buttons.

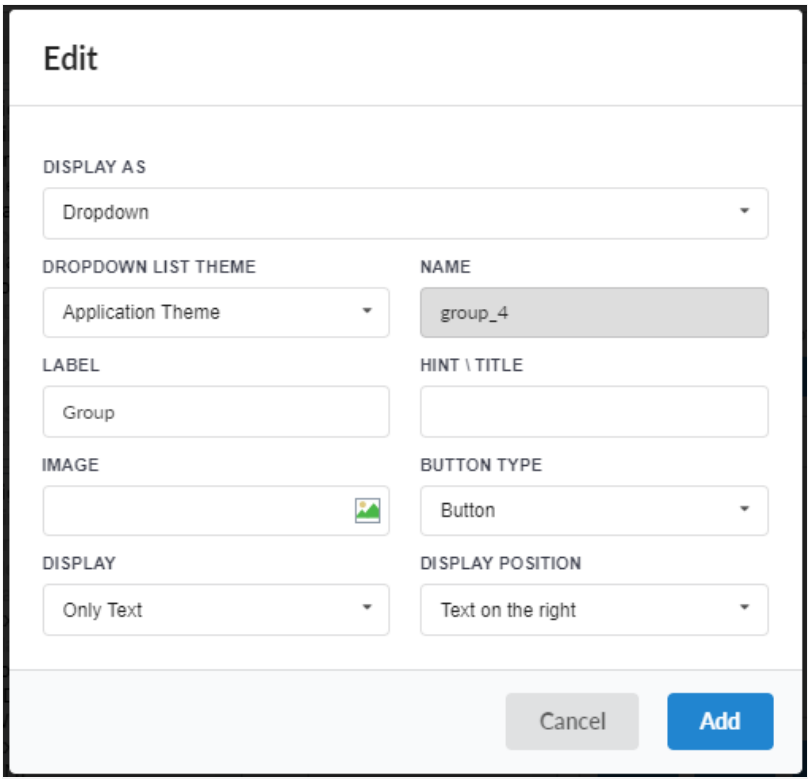
Edit

Edit an existent group of buttons.

Delete

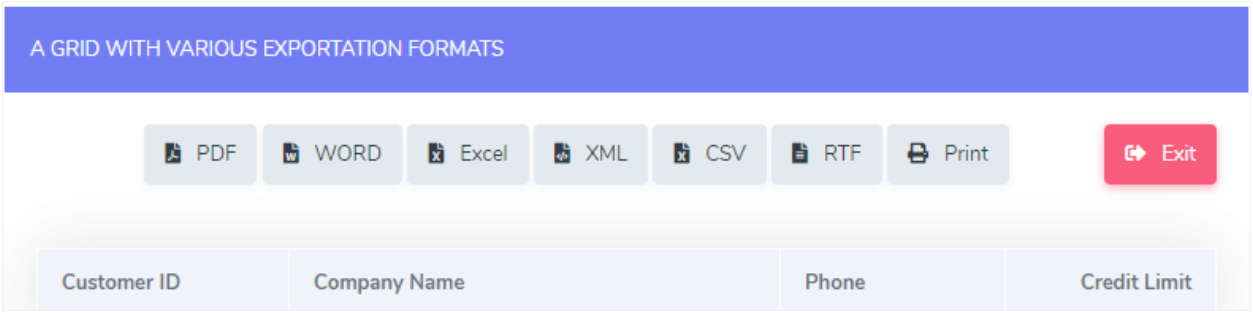
Delete the selected group of buttons.

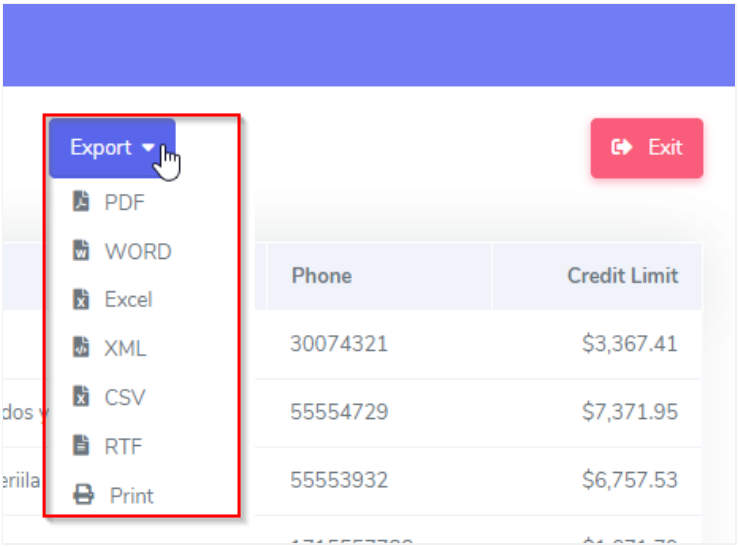
When you press the **Add** or **Edit** option, you can see the settings to configure the grouper:

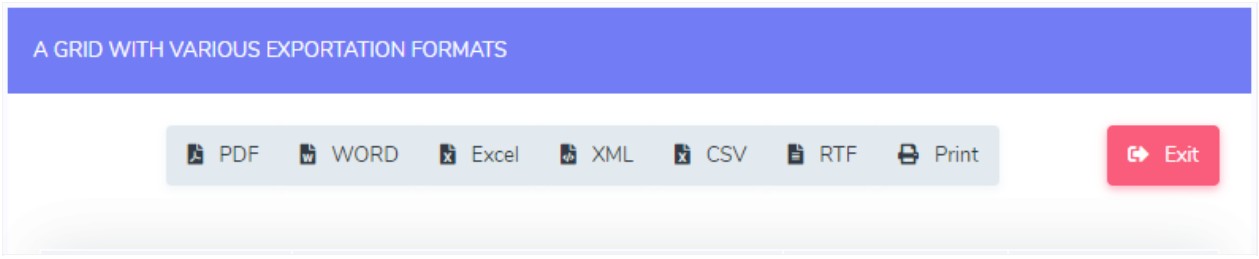


Display As

Allows displaying the group button as **Dropdown** or **Side by Side**.







Dropdown List Theme

Allows defining the Dropdown theme selecting between **Application theme** and **Button theme**.

Name

Allows defining a name for the button group.

Label

It is the displayed name for the button group in the application.

Hint\Title

Displays a hint to the end-user when the mouse is on the group of buttons.

Button Type

Allows displaying the button group as a Button, Image, or Link.

Image

Allows selecting an image for the button.

Display

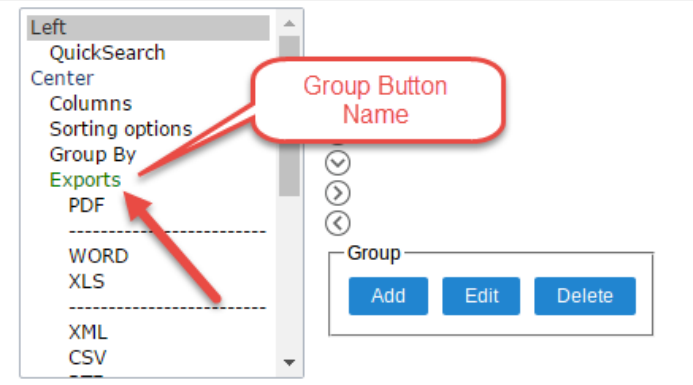
Defines if the button displays only Text, only image, or both.

Display Position

Defines the position of the Text or Image (Text to the right, Image to the right).

After creating a button group, you need to move the grouped buttons below of the Button Group and then move them to the

right. Like the image below:



Buttons Settings

Button	Label	Hint
QuickSearch		
Dynamic Search		
Insert		
Cancel		
Update		
Delete		
Exit		
Jump to		
First		
Previous		
Navigation by page		
Next		
Last		
Rows Counter		
New		

Button:

It displays the buttons available in the application.

Label:

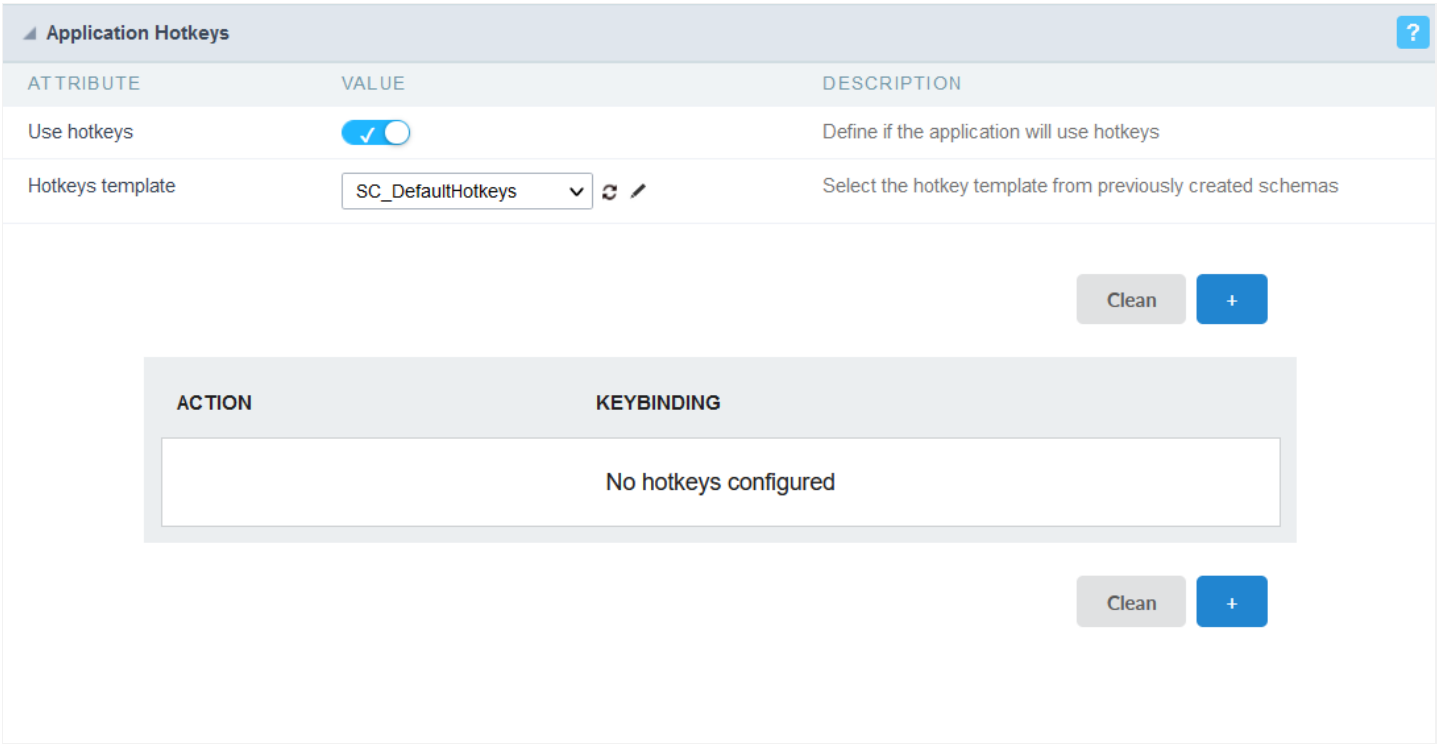
Allows defining the labels of the buttons to display for the users.

Hint:

Allows defining the buttons hint that to display for the users.

Application Hotkeys

Scriptcase allows creating shortcut keys to your applications. You can select a predefined template or create specific actions for an application.



Use hotkeys

Defines if the application uses hotkeys. When you enable this option, the default shortcut keys settings are disabled.

Hotkeys templat

Select the hotkey template previously created.

Action

Selects the triggered action when pressing the selected key.

Keybinding

Selects the keys responsible for executing the chosen action.

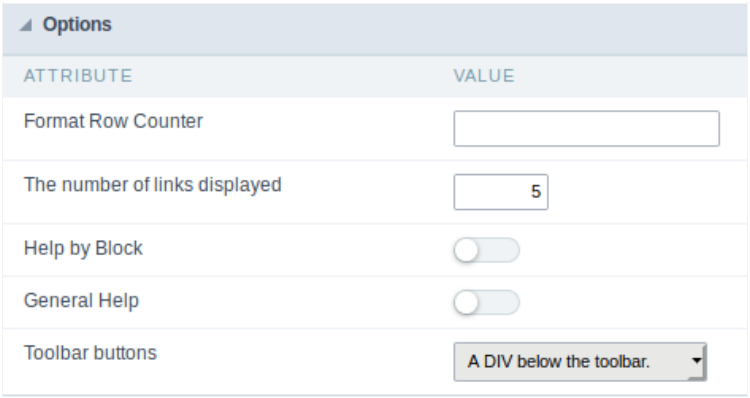
Add “+”

Adds a new action on the keys list.

Clear

It clears the selected hotkeys preference.

Options



Format Row Counter:

Allows defining the format of the row counter displayed on the application.

Example: (1 to 10 of 200)

The Number of Links Displayed:

Defines the number of links per page, when the navigation option is disabled.

Help by Block:

Indicates if the helps messages from relatives fields are grouped by block. In each field, we can define a help text. With this option activated, it shows up an icon in the block title bar to call the help page.

General Help:

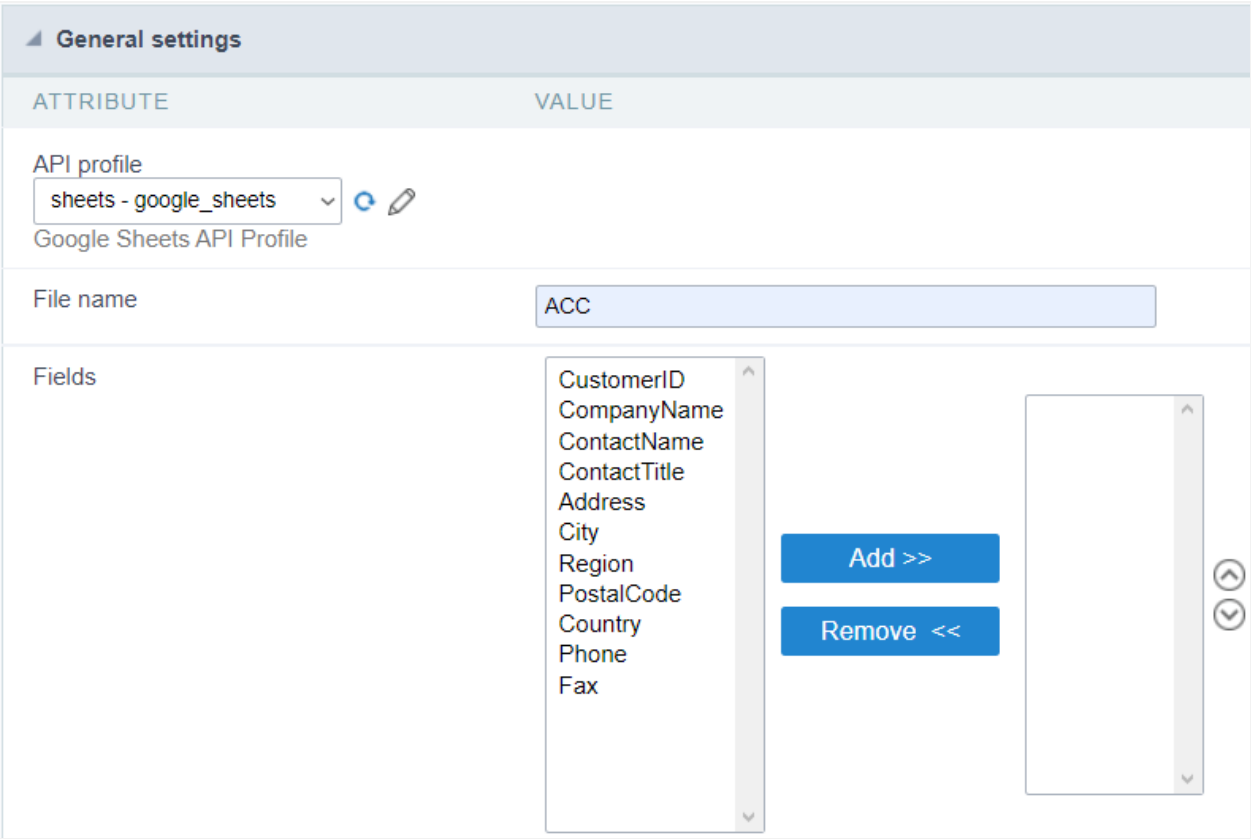
The General Help “consolidates” all the fields help pages in a single page, putting an icon in the toolbar to call the help page.

Related Link 🔗

Related Videos ▷

Form Export Settings

Google Sheets Settings



API profile

Allows you to set which api profile you want to use.

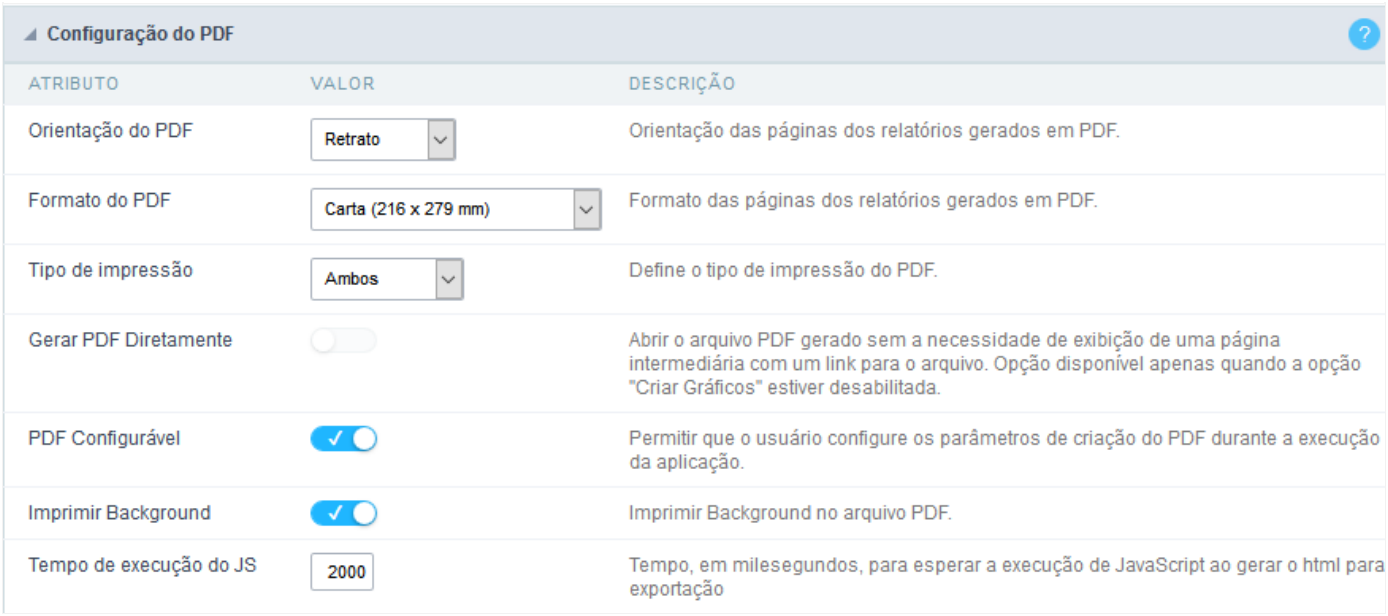
File name

Allows you to set a name to the file to be created or a file already created in Google Drive.

Fields

Allows you to choose the fields to be exported.

PDF Settings



PDF Orientation

Allows you to set whether to print in Portrait or Landscape orientation.

PDF Format

Allows you to define the type of form in which the PDF will be printed (letter, A4, etc).

Print Type

Allows you to set whether the print mode will be colored or economical.

Generate PDF directly

Opens the generated PDF file without the need to display an intermediate page with a link to the file.

Configurable PDF

Allows the user to configure the PDF creation parameters during the execution of the application.

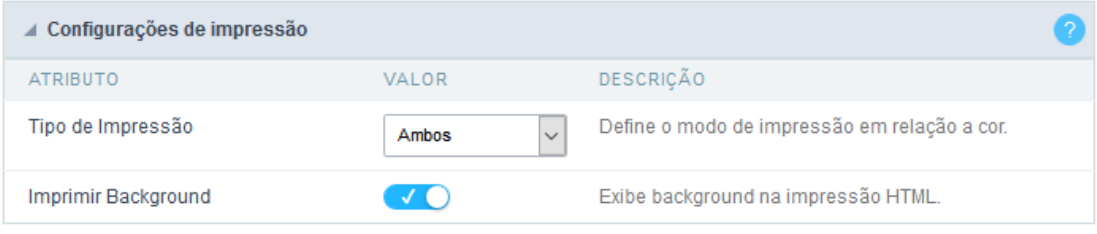
Print background

Allows you to print a background in the PDF file.

JS Runtime

Maximum time (in seconds) of server waiting when running JS.

Print Settings



Print Mode

Allows you to set the contents of the print file (Both, Current Page, Full Report).

Print background

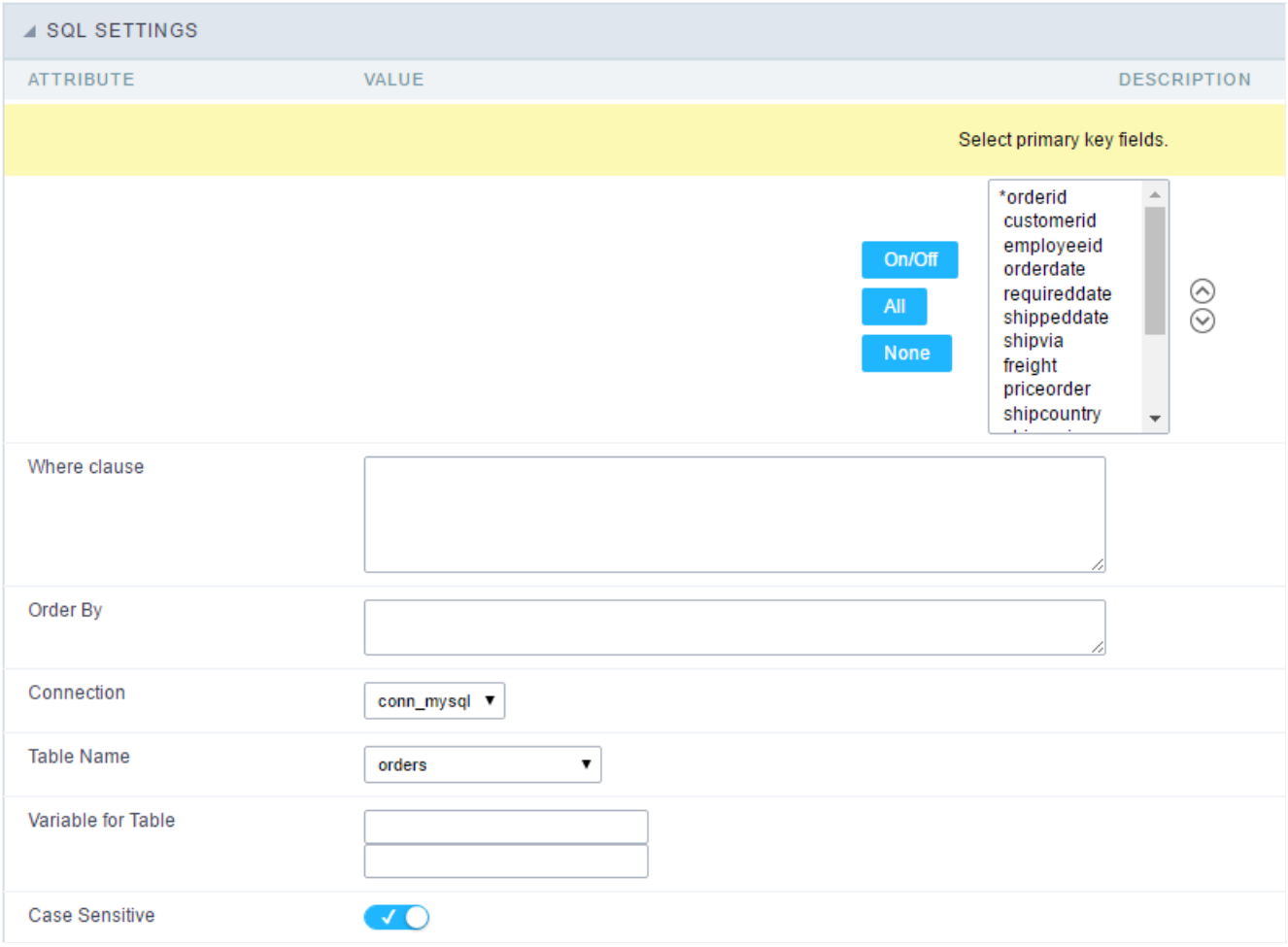
Lets you set whether the background will be displayed on the printout.

Related Link 🔗

Related Videos ▷

Fom SQL Settings

This interface allows configuring the related database settings, such as the Primary Key, Filters, Sorts.



SQL configuration Interface.

Select primary key fields

It lets you define the Primary key of the Form. ScriptCase already identifies Primary Keys, but you can manually inform or change it by using the buttons beside the fields list. See how the buttons work:

- **On/Off** : Adds or Removes the attribute that defines the primary key for the field. The primary key fields have an asterisk beside their names.
- **All** : Defines all fields as Primary Keys.
- **None** : Defines none fields as Primary Keys.
- **Sorting Button** : These are the arrows on the right side of the Combo box. It allows ordering the fields of the Primary Key, placing it in the desired order. To order them, click on the field and use the arrows to move it.

Where clause

It allows adding a WHERE clause to filter the SQL records.

Order By

It allows adding an ORDER BY clause to determine the order to display the records. By default, it uses the primary key to sort the records.

Connection

It allows defining the database connection of the application. You can change the connection to another one that has the same table.

Table Name

It informs the database table used in the Form.

Variable for Table

It allows to use a variable to change a part of the string containing the table name.

Variable for Table Configuration.

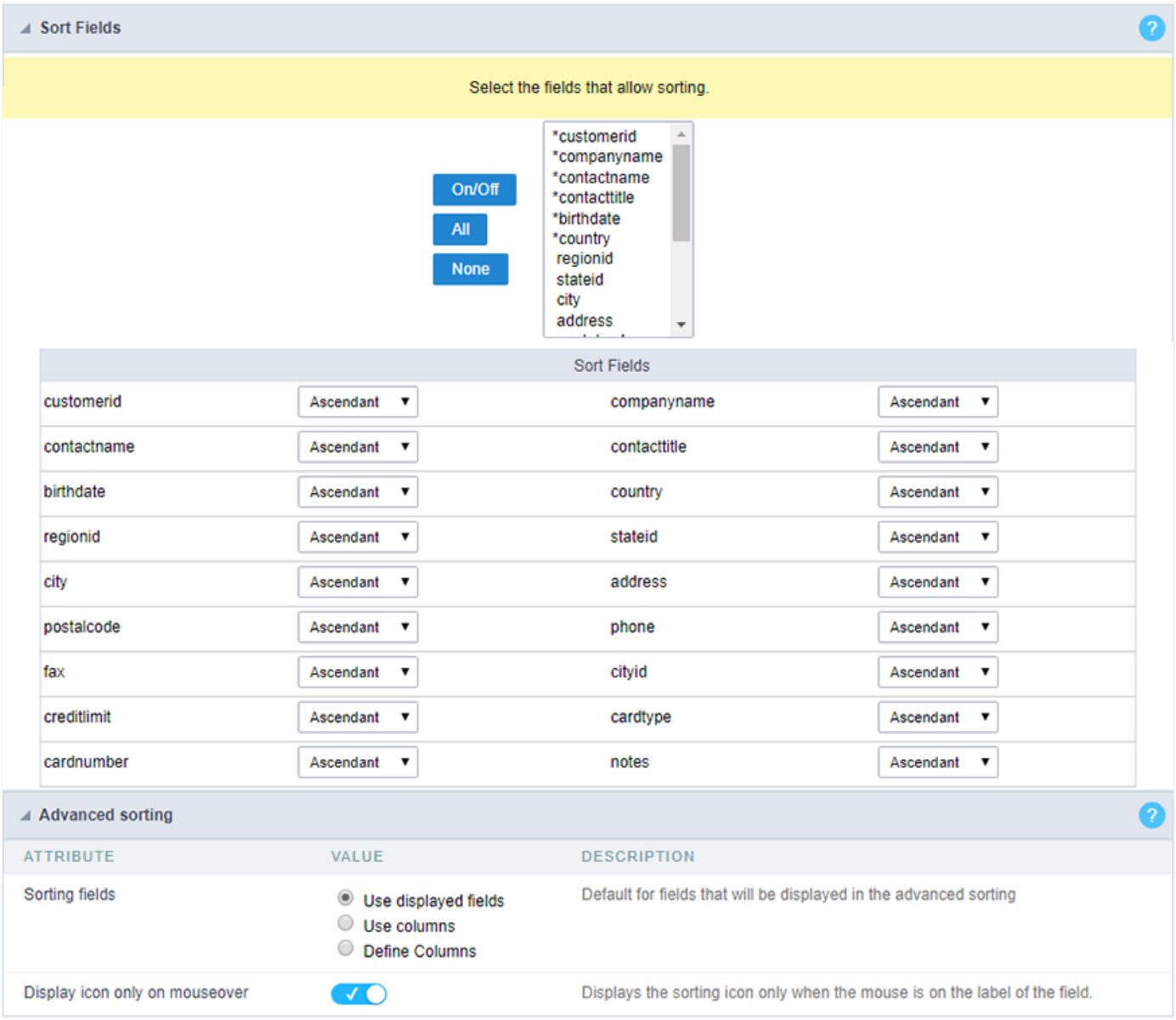
Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the table you want to replace (replace from).

Case sensitive

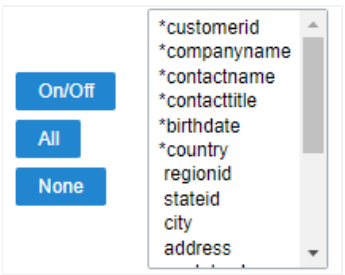
It defines if the database connection uses case sensitive or not.

Sorting

Field Settings



This property defines which fields allow the end-user to sort on the fly, just by clicking on the column (field) he wants to sort by. When sorting the fields, it preserves the GroupBys ordering by default. Also, Scriptcase enables the sorting of the first six application fields, but you can enable it for all fields.



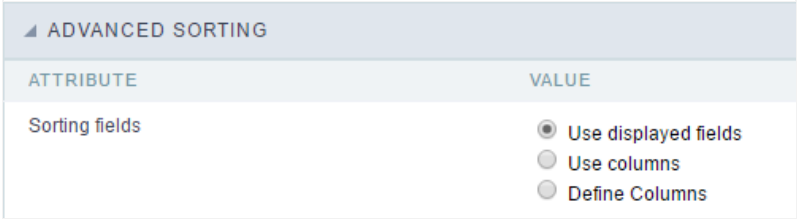
Sort Fields

It defines the sorting for each field of the application. You can set them as “Ascendant” or “Descendant”.

Sort Fields			
customerid	Ascendant	companyname	Ascendant
contactname	Ascendant	contacttitle	Ascendant
birthdate	Descendant	country	Ascendant
regionid	Descendant	stateid	Ascendant
city	Ascendant	address	Ascendant
postalcode	Ascendant	phone	Ascendant
fax	Ascendant	cityid	Descendant
creditlimit	Descendant	cardtype	Ascendant
cardnumber	Ascendant	notes	Ascendant

Advanced Sorting

Define the fields that are available when clicking on the “Sorting” button of the Grid.



Use displayed fields

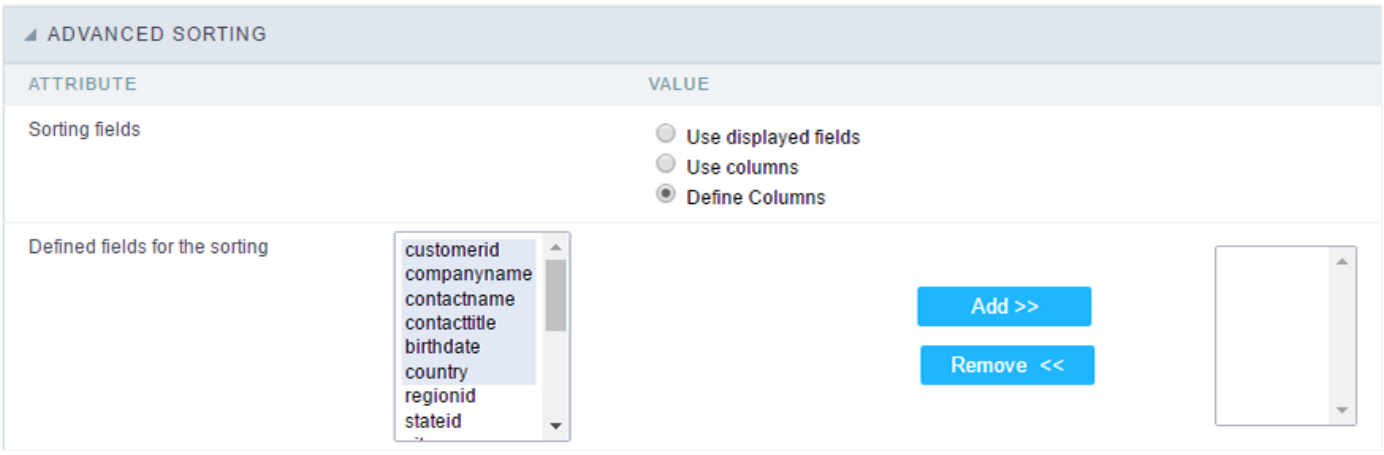
Defines that all the fields are available when clicking on the “Sorting” button of the application.

Use columns

Defines that all the fields set in the “Columns” option are available when clicking on the “Sorting” button.

Define Columns

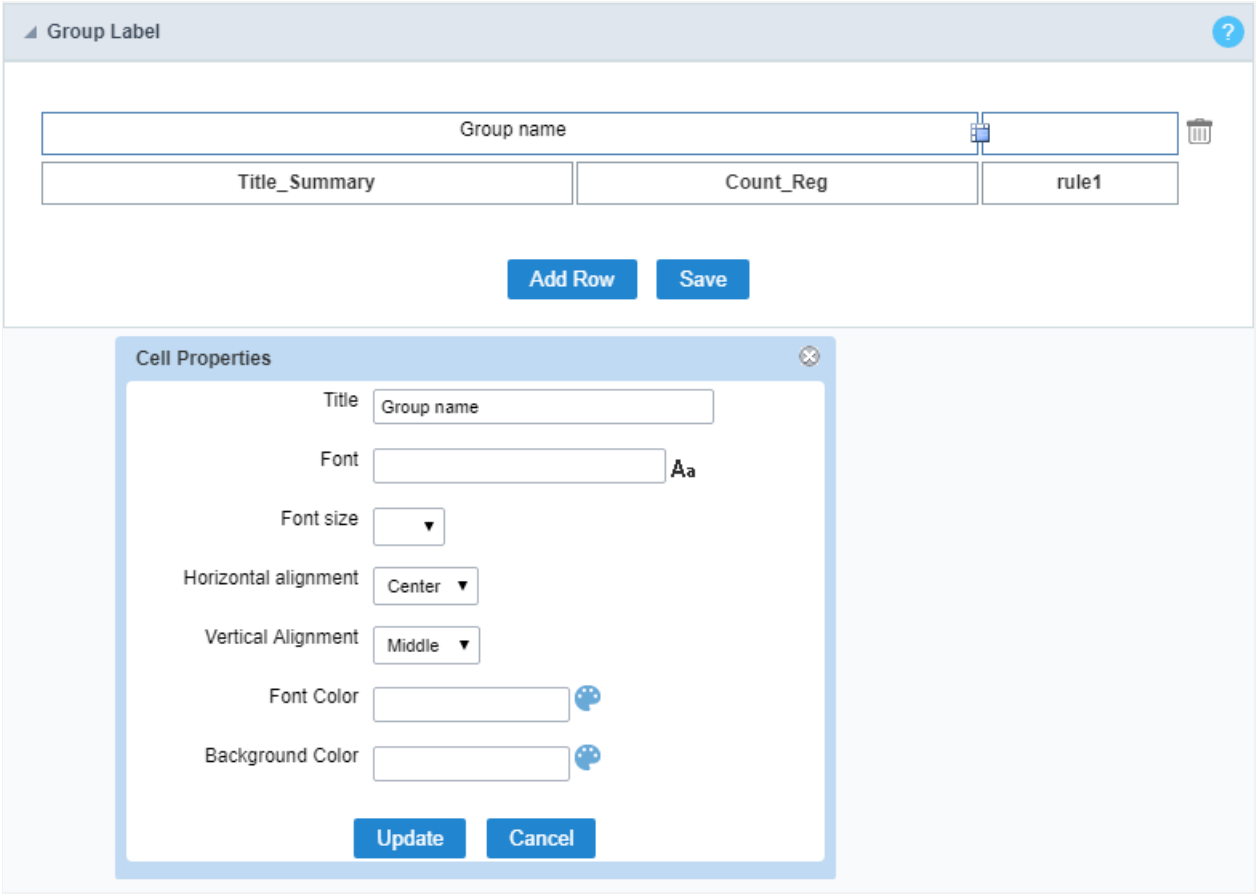
Allows selecting the fields freely.



Form Group Label

With this option, you can insert one or more titles for the columns in the application, allowing to modify its size, color, font, and position.

The Group Label doesn't work with a dynamic display of fields.



Editing Group Label cell properties

Title

It allows you to define a Title for the Group Label.

Font

Set the font family for the group label title.

Font Size

Set the font size for the group label title.

Horizontal Alignment

Define the horizontal alignment of the group label title. Left, Center, or Right.

Vertical Alignment

Define the vertical alignment of the group label title. Middle, Bottom, or Top.

Font Color

Set the font color for the group label title.

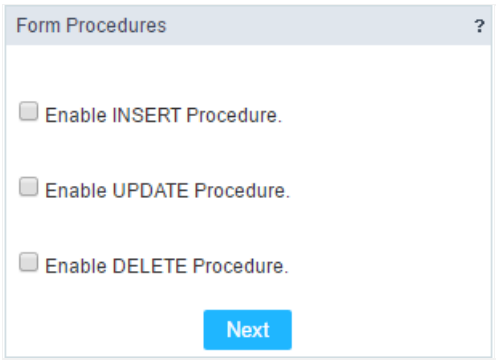
Background Color

Define a background color for the group label title.

Procedures

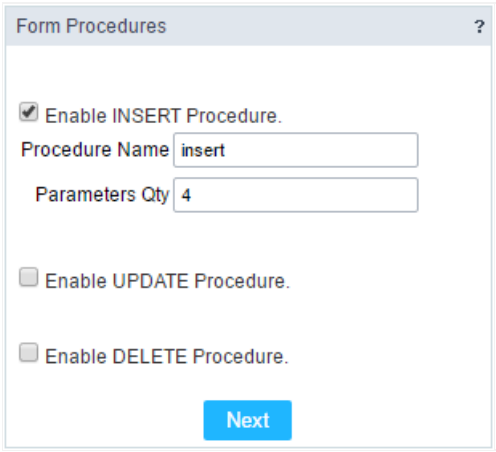
This interface allows configuring a Form application to execute the Stored Procedures from your database for Insert, Update and Delete records in the form. You don't need to use the three options simultaneously. If a Procedure option is not enabled, it continues to work as default by using the INSERT, UPDATE and DELETE commands.

See below, the step-by-step to enable a Stored Procedure for inserting records into the database.



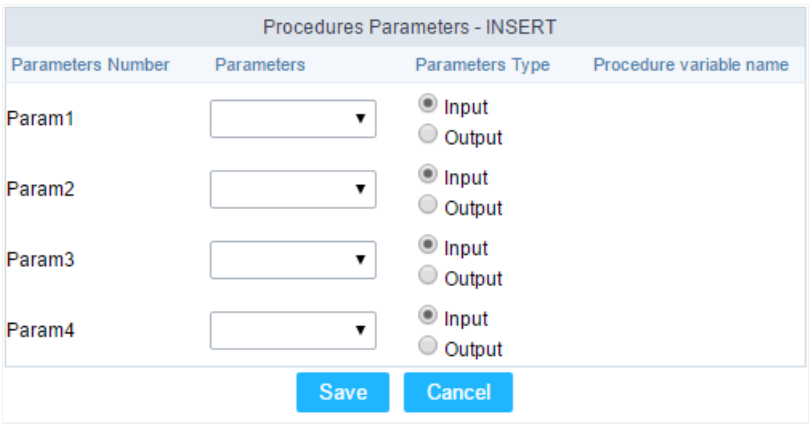
Form Stored Procedures Interface.

- Let's check the first option: "Enable INSERT Procedure". Then, inform the Name of the Procedure and the number of parameters.



Defining parameters for the Procedure.

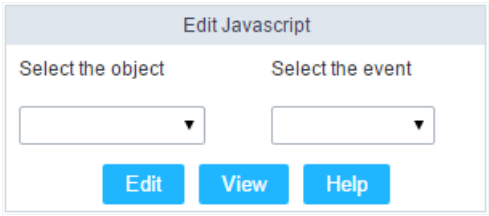
- Now associate the fields for each parameter of the Stored Procedure and the type of parameter (Input or Output). Click on the "Save" button to finish.



Passing the values to the Stored Procedure Parameters.

JavaScript

To coding with JavaScript in Scriptcase Form application, we must associate a JavaScript event to a form field.



Edit JavaScript Interface.

Select the object

This Combo box displays the fields of the form application, and also the form itself as an object

Select the event

Use it to associate the event with the selected field to apply the JS code. View the events available:

- **onclick**: Acts when clicking on the field.
- **onblur**: Acts when the focus leaves the object.
- **onChange**: Acts when the focus leaves the object, and there are changes in the value.
- **onFocus**: Runs when the focus gets in the object.
- **onMouseOver**: Runs when the mouse pointer hovers the object.
- **onMouseOut**: Runs when the mouse pointer moves out the object.

Events related to the Form

The events below are associated directly with the form object.

- **OnLoad**: This event runs when the page is loaded, also when clicking on the navigating buttons.
- **onSubmit**: This event runs when clicking on the “New”, “Save”, and “Delete” buttons.

Edit JavaScript

*Select the object and event, then click on the Edit button. It opens a page to inform custom JavaScript routines and standar

![Edit JavaScript Interface][javascript_edicao_code]
Edit JavaScript Interface

> The JavaScript language doesn't have the same behavior in all the browsers available. A tip is to test running the applicatic

- **OnClick Example**
  - When clicking on a field of type radio, you can enable or disable form fields according to the selected value.

```
if(document.F1.gender[0].checked){
   document.F1.maternity.disabled = false;
   document.F1.maternity.style.background='FFFFFF'
}
if(document.F1.gender[1].checked){
  document.F1.maternity.disabled = true;
  document.F1.maternity.style.background='FCEEBC'
}
```

To access the values of a radio field, you need to use the index.

- **OnBlur Example**

- You can define a warning for the field “weekly_work_time” when the focus is getting out it.

```
if (document.F1.tp_point[0].checked  && document.F1.weekly_work_time.value > '20')
{
    alert("The work time exceeds the limit allowed");
    document.F1.weekly_work_time.value = "";
    document.F1.weekly_work_time.focus();
}
```

- **OnChange Example**
  - By modifying the “Salary” of an employee and leaving the field, we’ll check if his “position” is “gardener”.

```
if (document.F1.salary.value > 2000.00 && document.F1.position.value == 'gardener'){

alert('When I grow up, I want to be a gardener);
}
```

- **OnFocus Example**
  - After informing the purchase value and selecting the payment method in a Select object “Select: pay_method”, the JavaScript code below calculates the value of the purchase.

```
if (document.F1.pay_method[document.F1.pay_method.selectedIndex].text == 'Money')
{
     document.F1.total.value = document.F1.pauchase_value.value;
}
```

- **OnMouseOver Example**
  - You can change the style (background color, font and font color) when the mouse hovers the field.

```
document.F1.field_name.style.bgColor = "0FFCCA"
```

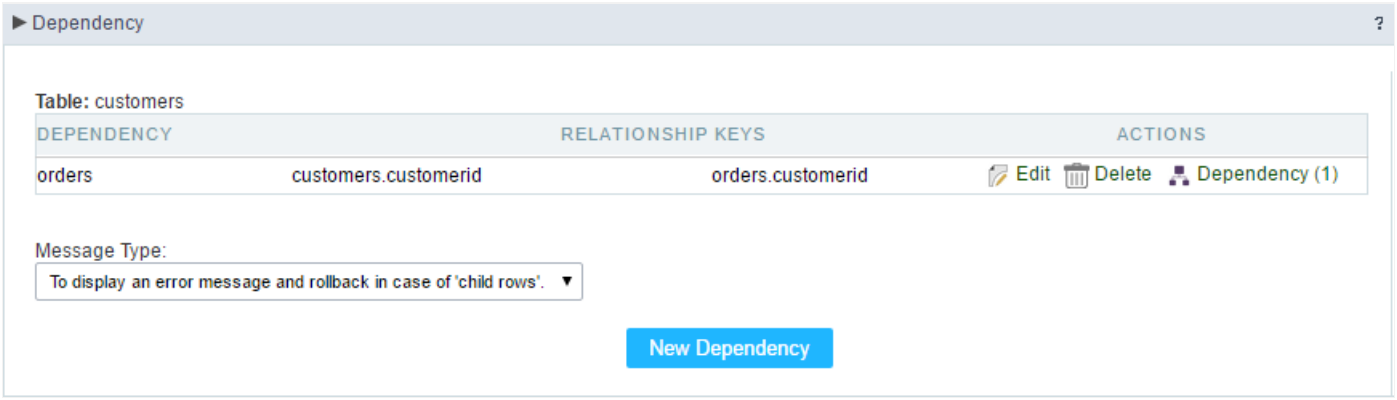
- **onMouseOut Example**
  - Sets the background color when the mouse moves out from the field.

```
document.F1.field_name.style.bgColor = "FFFFFF"
```

Dependencies

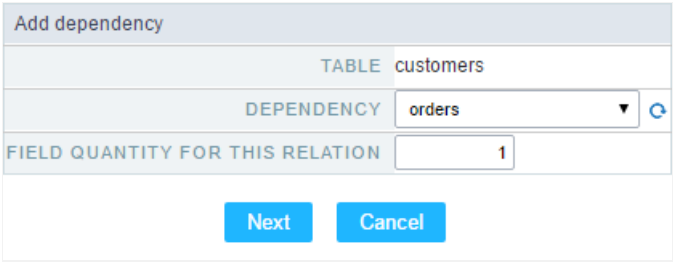
This feature allows linking the tables that contain relationships, dependencies, like the tables orders and order_details. This way, it's possible to delete a record in the table orders, and it automatically deletes all the details for that order as well. Below let's see a practical example of this feature.

1. Firstly, you must define a new dependency. Then you must select the dependent table. Click on the button New Dependency to start.



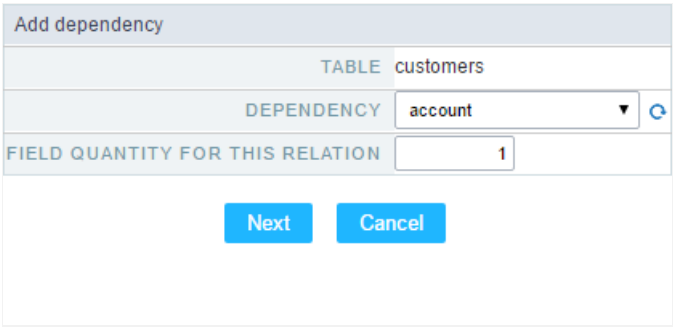
Creating a new Dependency.

1. You must inform the fields amount is related between both tables. In this example, we have only one field.



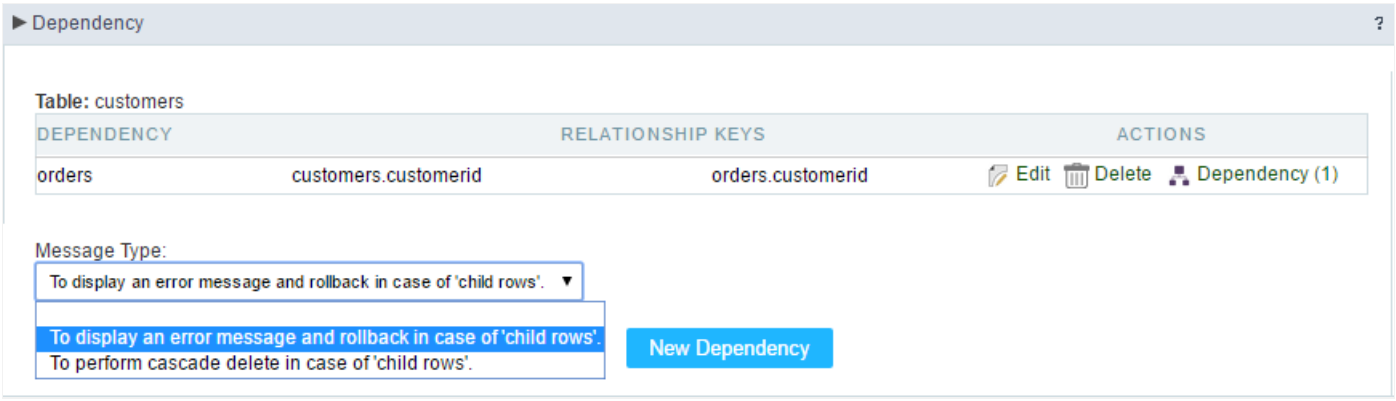
Dependency Table configuration.

1. Then select the related key fields: order_details->Orderid - orders->Orderid.



Dependency Key Fields Configurations.

1. Now we need to select behaviour for the application. Then click on the button Generate Scripts to define the Dependency Rule.



Configuring dependency Rules.

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application.

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Enable CSRF

With this option enabled, the scriptcase prevents a malicious attack on a page where unauthorized commands are transmitted through a user the page trusts.

These attacks are known as a “Cross-Site Request Forgery” attack.

Related Videos ▷ tem template

- Form overview
- Security Module (fundamentals)
- Security Module (learning SC from scratch)
- Log module (fundamentals)
- Log module (learning SC from scratch)
- Login Screen
- Control Module (macro sc_log_add)

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Related Link

- Security system template

Related Videos

- Form overview
- Security Module (fundamentals)
- Security Module (learning SC from scratch)
- Log module (fundamentals)
- Log module (learning SC from scratch)
- Login Screen
- Control Module (macro sc_log_add)

Headers

Disable XSS Auditor

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Related Link

- Security system template

Related Videos

- Form overview
- Security Module (fundamentals)
- Security Module (learning SC from scratch)
- Log module (fundamentals)
- Log module (learning SC from scratch)
- Login Screen
- Control Module (macro sc_log_add)

Captcha

Captcha

Use Captcha

Defines if the application uses Captcha.

Number of Characters

Amount of characters in the Captcha image.

Character List

List of characters used in the Captcha.

Label

The message displayed for the Captcha.

Error message

Captcha error message.

Height

Height of the Captha image. (in pixels)

Width

Width of the Captha image. (in pixels)

Font Size

Font Size of the Captha image. (in pixels)

Reload

Display the refresh button in the Captcha.

Select one layout

It offers layouts for display the Captcha.

Recaptcha

ReCAPTCHA is an API provided by Google for forms. It adds security, preventing automatic submission of forms through robots.

reCAPTCHA sample:

1. First, we must request an API Key to activate reCAPTCHA into a Scriptcase application by following the steps below:

   To get a **Site key** and **Secret Key** go to the link: https://www.google.com/recaptcha/admin#list.
   See the image:

Label

It is a project identifier to create the reCAPTCHA keys.

Choose the type of reCaptcha

We must choose the option **reCAPTCHA V2**.

Domains

We can insert multiple domains (one per line) to limit the API uses.

1. Then, we need to accept the Terms of Service (“Accept the reCAPTCHA Terms of Service”).
2. When clicking on **Register**, the page refreshes and shows the integration of reCAPTCHA information. There we can get the **Site Key** and **Secret Key**:

1. Now, we can set the Scriptcase application security:

Site Key

Key generated by Google after reCAPTCHA project creation.

Secret Key

Key generated by Google after reCAPTCHA project creation.

Theme

Defines the reCAPTCHA color. There are two options:

- **Light** :
- **Dark** :

Type

The type of reCAPTCHA. There are two options:

- **Audio**:
- **Image**:

Size

The size of reCAPTCHA. There are two options:

- **Normal** :
- **Compact**:

Position

Here we can define the reCAPTCHA component alignment:

- **Left**: Position the reCAPTCHA component at the left.
- **Center**: Position the reCAPTCHA component at the center.
- **Right** : Position the reCAPTCHA component at the right.

Related Link

- Security system template

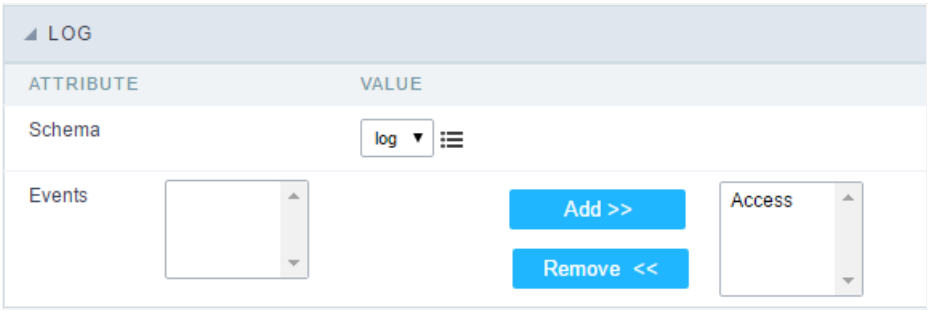
Related Videos

- Form overview
- Security Module (fundamentals)
- Security Module (learning SC from scratch)
- Log module (fundamentals)
- Log module (learning SC from scratch)
- Login Screen
- Control Module (macro sc_log_add)

Form Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Related Link

Related Videos

Form - Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

How to synchronize table fields with a form

Anytime you add/delete a field or change its data type, you have to synchronize your form application with the connected table to apply that changes.

For more information, click here.

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid		
Procedure		
Chart		
Form		
Control		
Calendar		

Application Table Fields Virtual Fields

Search

PDF Report

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven't manage fields.

Related Links

- Data entry marks
- Unique Key Validation
- Autocomplete
- Digital Signature
- Watermark
- Form preview
- Form layout
- Data validation
- Combobox with data edit form
- Working with Combobox 1/2
- Working with Combobox 2/2
- How to save uploaded files to Google Drive
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Related Videos

- Form types
- Upload files to cloud
- Upload of Multiple Images
- Editor HTML
- Upload and multi upload fields
- Multiselect (google select field)
- How to save uploaded files to Google Drive

Form - Text Field

General Settings

This type of field allows the developer to create quickly inputs to insert and update data, where the final user can inform its data to be allocated in its database.

Type of Data

Defines the type of the field in the application. In this case we should select **Text**.

Label

Defines the title that will be displayed in the field when the application is executed. The terminology used in the interface is fundamental to the system have a good usability, whe should use names and familiar terms to the final user of the application, instead of using terms from the system.

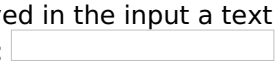
For example, this text field that has the name **customerid**, the client would have a much better understanding of the functionality of the field when we define the label as **Customer Name**.

Not only a fixed text, the **Label** attribute allows the use of langs to define the field's title, making it possible to use your application in a multi language project.

Label below field

Defines the message to be displayed below the field.

Watermark

Informing a text to the **Watermark** it will be displayed in the input a text as an example that can be informed in the field. The result after set will be this: 

Initial Value(type)

Allows the initial definition to the field when the form is in insert mode. It is possible to choose between two options:

Defined Value: When this option is selected, the Initial Value attribute will be available, where we should inform the field's initial value. For example, my initial value is **Arlindo**, when a new register is inserted, the field **Seller Name** will be initialized as Arlindo.

System Data: When this option is selected, the initial value will be the actual date of your computer's system.

Amount of Characters

Allows to set the width of the text field's input that varies with the amount of characters informed. Although, if the amount of characters typed are greater than the setting, the text will be pushed to the left, to keep the maximum amount of characters as defined.

Show HTML Content

When this option is active every HTML, CSS and JavaScript content that are in the database will be displayed with the main value.

Validation Image

When this option is active, a image will be displayed next to the informed field if the field is according to the settings defined by the developer.

In the example below, the field was set to receive at least 5 characters, see what happens when informed

only 4 characters:

However if informed 5 or more characters the field will be displayed as:

Password Field

When this option is active, the text field will be converted to the format used in password fields. For example:

View password characters

By enabling this option, a button will be displayed in the password field so that the password is displayed when clicked.

Save Variable

Allows to save a session variable(global variable) with the field's value, to be used in others applications.

For example, in the login form the username can be saved in session and displayed on the header of others applications.

Variable Name

In this attribute we should define the name of the session variable, active in the previous item, that will receive the field's value.

We should inform only the variable's name, - **var_rating**.

The method to use its value is global variable.

Field Mask

Defines the field mask. There are two types of mask described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

For example, it is possible to set the mast to display a telephone number:

It will be show with this format on runtime:

It is also possible to set the field mask like those examples:

Field mask examples:

Telephone number

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678

Field	Mask Input	Typed Value	Formatted Value
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Validate with mask

When this option is active it is possible to insert the data with the mask set in the **Field Mask** option.

Complete to the left

When this option is used, the value defined will be added on the left of the information inserted in the field if the value is lower than the maximum set in the **Maximum Size** option.

Field size in database

Defines the field's size related to the size set in the database. This value is already set automatically by default when the application is generated.

Hidden Field

When this option is active, the field will be hidden in the application on runtime.

Label Field

When this option is active, the field will be altered to only a label where the info will be displayed, where updates or inserts will not be possible.

Save HTML tags

When this option is active, it allows to HTML tags in the field to be inserted with the data, instead of being interpreted.

Text input in JavaScript

When this option is active, it will be show every JavaScript content inserted in the database with its main value.

> This option can only be used using the **Editable grid view**.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disable Field

Allows the developer to disable the field, so the user can not type a value according with the option defined by the developer.

The available options are: **No** - Initial value of the attribute, this option does not disable the field.
Update Mode - This option only disables the field when editing existing registers.
Insert Mode - This option only disables the field when inserting new registers.
Insert / Update Mode - This option disables the field when editing exciting registers or when inserting new registers.

HTML Type

HTML object used to display the field in the form.

SQL Type

Informs the type of the field in the database.

Values Format

 Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed.

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To

include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Multiple Lines Text Field

General Settings

Multiple Lines Text field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Multiple Lines Text , you can inform a Text value to the field in multiple lines.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : you will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Define the amount of characters allowed for the field.
- **Validation Image** : Allows to display an image next to the field when it is being validated.
- **Lines** : Allows to define the amount of lines that the field will have at start.
- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the field size in bytes. it is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Save HTML tags** : This option allows to save the HTML content of the field to the database.
- **Text input in JavaScript** : Allows to save JavaScript code informed in the field to the database.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed.

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

```
SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
```

The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position

(left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Data entry marks
- Validation

Related Videos ▷

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Integer Field

General Settings

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Label:

It sets the title that will be displayed in the field when you run the application. The terminology used in the interface is fundamental for your system has a good usability. In this case, we recommend you to use names and terms familiar to the end user of the application, instead of using terms originated in the system.

Label below field

Defines the message to be displayed below the field.

Watermark

Informing a text for the **watermark** it will be displayed on the input text with an example of what can be reported in the field. The result after that will be configured:

Initial value (type)

Allows setting an initial value for the field when the application is in inclusion. You can choose between two options:

Pre-set value: When you select this option, the Initial Value attribute will be made available in the place where we inform the initial value of the field.
For example, my initial value is **Arlindo** when inserting a new record the **seller_name** field will start with Arlindo.

System date: When you select this option, the initial value will be the current system date of your computer.

> If you select the type system date, it will be not necessary to fill in the initial value attribute.

Amount of characters

It allows you to set the width of the input text field according to the amount of characters. However, if the quantity entered is greater than the set for the characters, the text will be pushed to the left, in order to ensure the maximum amount of characters set in the option of **Values formatting**.

Validation Image

When you enable this option, an image will be shown next to the field informing whether the field is in accordance with the settings of *minimum size* and *Maximum* size (as you can see in the images below) defined in the Formatting value option.

In the example below, the field was set to receive at least 5 characters, see what happens with the image when it receives only 4 characters:

However, if the value inserted has 5 or more characters the the image will change according to example bellow:

Use slider:

It displays a slider component in the field. So you can increase or decrease the value sliding the cursor. You can also customize the increment value, if it increments the value 1 by 1, 2, 5, 10... N.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Validate with Mask

By enabling this option you will be able to enter data according to the mask that was configured in the option **Field Mask** and Scriptcase will validate it.

Record Variable

Allows you to record a session variable with the value of the field ([global variable] [var_glob]), to be used in other applications.

Example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable Name

In this option you must set the session variable name, enabled in the previous item, which will receive the value of the field.

You need to inform only the variable name, for example: **var_seller**.

The recovery of the value is made as [Global Variable][var_glob]{:target='blank'}.

Field size in Database

This option sets the size of the field relative to the size that is configured in the database. By default this value is already configured automatically when the application is generated.

Hidden Field

This option when enabled will hide the field inside the application at the time of execution.

Label Field

By enabling this option, the field will be changed to only one label where the information will be displayed, so it is not possible to make changes or inserts in the field configured as label.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disable field

Allows you to disable the field, making it impossible for the user to enter a value according to the developer-defined option.

The available options are:

No - initial value of the attribute, this option does not disable the field. **Update mode** - This option disables the field only when editing the records. **Insert mode** - This option disables the field only inserting new records. **Update/Insert mode** - This option disables the field in both editing and inserting new records.

HTML Type

HTML object used to display the field in the application.

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Database Value

It allows you to define how the value to be written to the database.

Preset value:

It allows you to specify a fixed value or associate it to global variables created inside the project and save it automatically to the database field.

Auto increment (automatic):

It allows you to use the increase generated automatically by the database to fill the field. It is used only when the database field is of type AUTO INCREMENT or similar. For the databases that use string, such as Oracle, PostGres and Firebird, you must inform the name of the string.

Auto increment (manual):

The application itself will simulate an auto increment in the field. So the value of the field will be calculated automatically by the application.

Date of inclusion:

The field will be populated automatically with server date during the inclusion.

Date of update:

The field will be populated automatically with server date during the update.

User IP:

The field will get the IP of the machine that is accessing the application.

Calculated by the database:

The field value is assigned by the database itself. The field will not be used in the process of updating. For example: it will be updated via a Trigger.

Null:

The field will be automatically filled with null.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:

Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

```
SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}
```

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1

Label	Value	Start	Size
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.

- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be

displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Data entry marks
- Validation

Related Videos ▷

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Decimal Field

General Settings

Decimal field configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser’s autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser’s autocomplete.
- When **enabled**, the field’s autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit grouping

It defines if the field displays the digits separator.

Maximum Size

It defines the max size of the field.

Minimum Value

It defines the min value of the field.

Maximum Value

It defines the max value of the field.

Decimal Precision

It defines the number of decimal places for the field.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Accept

It determines if the field accepts only negative, positive, or both numbers.

Show a calculator

It displays an icon beside the field to help the user calculating the field value.

Auto-complete with zeros

It automatically informs the decimal places when not specified. If disabled, the user always needs to inform the decimal value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected

below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit

- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Currency Field

General Settings

Currency field configuration Interface.

Data Type

Define the field type to Currency. It allows defining the currency number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser’s autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser’s autocomplete.
- When **enabled**, the field’s autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit grouping

It defines if the field displays the digits separator.

Currency symbol usage

It defines if the field displays the Currency Symbol of the Regional Settings.

Maximum Size

It defines the max size of the field.

Minimum Value

It defines the min value of the field.

Maximum Value

It defines the max value of the field.

Decimal Precision

It defines the number of decimal places for the field.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Accept

It determines if the field accepts only negative, positive, or both numbers.

Show a calculator

It displays an icon beside the field to help the user calculating the field value.

Auto-complete with zeros

It automatically informs the decimal places when not specified. If disabled, the user always needs to inform the decimal value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The

select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Date Field

General Settings

Date field configuration Interface.

Data Type

Define the type of field. When setting it to Date, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server Date.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in "Insert Mode", "Update Mode" or in "Insert and Update Mode".

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Date separator

It allows you to inform the separator symbol for the date.

First Day

Defines the first day of the week.

Display

It offers predefined formats for displaying dates.

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

Use Combo-box

It allows you to select the date from a combo-box.

Minimum Date

- **Fixed Date**: Inform the minimum date allowed presented in the selected date format.
- **Current Date**: By clicking on the icon next to the text field, ScriptCase offers the following options:
  1. **Actual date**: Set the current date as the minimum allowed date.
  2. **Actual date with increment**: Set the minimum date with the current date more(+) the days, months, or years you want to increment.
  3. **Actual date with decrement**: Set the minimum date with the current date minus(-) the days, months, or years you want to decrement.

Maximum Date

- **Fixed Date**: Inform the maximum date allowed presented in the selected date format.
- **Current Date**: By clicking on the icon next to the text field, ScriptCase offers the following options:
  1. **Actual date**: Set the current date as the maximum allowed date.
  2. **Actual date with increment**: Set the maximum date with the current date more(+) the days, months, or years you want to increment.

3. **Actual date with decrement**: Set the maximum date with the current date minus(-) the days, months, or years you want to decrement.

Display Format

It enables the Date format beside the field when informing the date.

Date display position

Available when activating the **Display Format** flag, defines where the filling mode will be displayed. The options are: **Right**, **B Lower** and **Watermark**.

Display Calendar

It enables a calendar icon beside the field that allows selecting the date from a calendar to fill the input.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Data validation
- Macro sc_time_diff
- Macro sc_date_empty
- Macro sc_date_dif_2
- Macro sc_date_dif
- Macro sc_date_conv
- Macro sc_date

Form - Time Field

General Settings

Time field configuration Interface.

Data Type

Define the type of field. When setting it to Time, you can inform a time format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in "Insert Mode", "Update Mode" or in "Insert and Update Mode".

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the Time. When not enabled, it displays the Time separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying Time.

Minimum time

Minimum value accepted by the field.

- **Fixed Time** : Enter the minimum time that scriptcase will criticize in the time type field in the format as presented.
    - **Current Time** : When clicking on the icon next to the box, scriptcase provides the following options:
    - **Current Time with Increment** : The minimum time will be the current time (+) the days or months or years that you increment.
    - **Current Time with Decrement** : The minimum time will be the current time (-) the days or months or years that you want to decrement.

Maximum time

Maximum value accepted by the field.

- **Fixed Time**: Manually enter the maximum time that scriptcase will input in the time type field in the format as it is presented.
    - **Current Time**: When you click the icon next to the box, scriptcase provides the following options:
    - **Current Time with Increment**: The maximum time will be the current time (+) the days or months or years that you increment.
    - **Current Time with Decrement**: The maximum time will be the current time (-) the days or months or years that you want to decrement.

Display Format

It enables the Time format beside the field when informing the time.

Display position

Available when activating the **Display Format** flag, defines where the filling mode will be displayed. The options are: **Right**, **B Lower** and **Watermark**.

Display full name

Displays the full name instead of the mask. Ex: day month year.

Use Time picker

It uses the JQuery plug-in to choose the time.

Database Value

Allows to define a value to be saved to the database.

 Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Data entry marks
- Validation

Related Videos ▷

- Form field types
- Form overview
- Data validation
- Macro sc_time_diff
- Macro sc_date_empty
- Macro sc_date_dif_2
- Macro sc_date_dif

- Macro sc_date_conv
- Macro sc_date

Form - Date and Hour Field

General Settings

Date and Time field configuration Interface.

Data Type

Define the type of field. When setting it to Date and Time, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server Date and Time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in "Insert Mode", "Update Mode" or in "Insert and Update Mode".

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Regional Settings

Allows you to apply regional date formatting settings to the field. When not selected, the Date Separator and Time Separator attributes will be displayed.

Display

Allows you to select the display format of the date/time field.

Minimum Date

Minimum value accepted as input for the field.

- **Minimum Date** :
  - **Fixed Date** : Enter the minimum date that scriptcase will criticize in the date type field in the format as presented.
  - **Current Date** : When clicking on the icon next to the box, scriptcase provides the following options:
  - **Simple Current Date** : Will leave the current date as the maximum date, that is, no one born after the current day will enter the form.
  - **Current Date with Increment** : The minimum date will be the current date (+) the days or months or years that you want to increment.
  - **Current Date with Decrement** : The minimum date will be the current date (-) the days or months or years that you want to decrement.

Maximum Date

Maximum value accepted as input for the field.

- **Maximum Date** :
  - **Fixed Date** : Enter the maximum date manually that scriptcase will criticize in the date type field in the format as it is presented.
  - **Current Date** : When clicking the icon next to the box, scriptcase provides the following options:
  - **Simple Current Date** : Will leave the current date as the maximum date, that is, no one born after the current day will enter the form.
  - **Current Date with Increment** : The maximum date will be the current date (+) the days or months or years that you want to increment.
  - **Current Date with Decrement** : The maximum date will be the current date (-) the days or months or years that you want to decrement.
  - **Display Format** : Enables display of the mode

Minimum time

Maximum value accepted by the field.

- **Fixed Time**: Manually enter the maximum time that scriptcase will input in the time type field in the format as it is presented.
  - **Current Time**: When you click the icon next to the box, scriptcase provides the following options:
  - **Current Time with Increment**: The maximum time will be the current time (+) the days or months or years that you increment.
  - **Current Time with Decrement**: The maximum time will be the current time (-) the days or months or years that you want to decrement.

Maximum time

Maximum value accepted by the field.

- **Fixed Time**: Manually enter the maximum time that scriptcase will input in the time type field in the format as it is presented.
  - **Current Time**: When you click the icon next to the box, scriptcase provides the following options:
  - **Current Time with Increment**: The maximum time will be the current time (+) the days or months or years that you increment.
  - **Current Time with Decrement**: The maximum time will be the current time (-) the days or months or years that you want to decrement.

Hour value

Sets the value to be assigned to the time when it is empty and a date is selected in the date picker.

- **Available values are**: Start of day, End of day and Current time.

Apply date limits to the calendar

Apply Minimum and Maximum Date limits when selecting calendar dates, avoiding the selection of dates outside these limits.

Display Format

Display the field format in the data cell.

Date display position

Available when activating the **Display Format** flag, defines where the fill mode will be displayed. The options are: **Right**, **Below** and **Watermark**.

Display full name

Displays the full name instead of the mask. Ex: day month year.

Group Date and Time

Group date and time in the same field.

Display Calendar

Allows you to display a calendar icon next to the field. This allows you to select the month and year, passing the value defined in the calendar to the field.

New Calendar

Show a new calendar with jquery, or the old calendar.

Years Limit

Number of years that will be displayed on the calendar.

View week number

Display week number in application.

Additional months

Display additional months in the calendar.

Available options are: 1, 2 and 3.

Show Combo year and month

Show Combo Year and month in calendar

Database Value

Allows to define a value to be saved to the database.

 Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks

Related Videos

- Digital Signature

Related Links

Related Videos

- How to save uploaded file to Google Drive
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_field_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record
- Macro sc_date_dif
- Macro sc_date_conv
- Macro sc_date

Form - HTML Image Field

General Settings

HTML Image field Configuration Interface.

Data Type

Define the type of field. When setting it to HTML Image, it allows to display an image into the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Image

Set it to display an image. The icon “Select Image” lists all images from Scriptcase and your uploaded images. The “Upload an image” option allows you to send a copy to the Scriptcase server.

Border

Define the width of the Image border in Pixels.

Width

Define the image width size in Pixels.

Height

Define the image height size in Pixels.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.

- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is

being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Label Field

General Settings

Label field Configuration Interface.

Data Type

Define the type of field. When setting it to label, it displays the value with no input to change it.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Text

You can inform a text to be displayed beside the field.

Reload

It links the informed text to reload the form when clicking on it.

Position

It allows positioning the field in the block.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.

- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Card Field

General Settings

Credit Card Number Configuration Interface.

- **Data Type** : Defines the type of field for the application. When it is defined as a Credit Card Number, the field verifies if the value is valid.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Creates a placeholder on the field with the text informed.
- **Initial Value (type)** : Set the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Defines the amount of characters allowed for the field.
- **Validation Image** : Display an image next to the field when it is being validated.
- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to determine the max amount of characters to be typed in.
- **Field Mask** : Defines the display mask for the field. There are three types of masks that can be merged.

Character	Description
9	Represents a numeric character (0-9)
A	Represents an alpha numeric character (A-Z,a-z)
*	Represents any alpha numeric character (A-Z,a-z,0-9) typed by the user.

It is possible to merge two or more masks simultaneously, separated by a semi coma with the smallest mask at start. The replacement occurs when the user is typing when the lowest amount of character exceeds.

Examples of Masks

Field	Mask	Informed Value	Formatted Value
Telephone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Software Key (Only Numbers)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters an Numbers)	**---**	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plates	AAA - 9999	QWE1234	QWE - 1234
ScriptCase License	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple Masks (Telephone)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

- **Hidden Field** : This option makes the field hidden, but still allows it is value to be processed through JavaScript or PHP.
- **Label Field** : This option make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Display the data type of field in the database.

Database Value

Allows to define a value to be saved to the database.

 Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.

- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Unique Key Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Card Type Field

General Settings

Credit Card field configuration Interface.

Data Type

Define the field type to Credit Card. It sets a combo-box for selecting the credit card type.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Field size in database

It determines the field size in bytes. It limits the max quantity of the allowed characters.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

 Label Configuration Interface.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn't chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn't chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn't chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Database Value

Allows to define a value to be saved to the database.

 Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookups Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

```
SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
```

The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position

(left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Data entry marks
- Validation

Related Videos ▷

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - E-mail Field

General Settings

Email field configuration Interface.

- **Data Type** : Defines the type of field for the application. When set to Email, the field applies validations for an email format.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Defines the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.

- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Defines the amount of characters allowed for the field.
- **Validation Image** : Displays an image next to the field when it is being validated.
- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Defines the name for the session variable that will receive the field value.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through Javascript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Display Icon** : Displays the Email icon next to the field.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - URL Field

General Settings

URL field configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Display Icon

Displays an icon beside the field.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

Show the URL as a clickable link

It sets the content of the field as a clickable link.

Target handling where the link will open

Target to open the link when clicking on it.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only

when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**

- **Font** : Allows to choose the font type, that will be applied to the application field.
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - HTML Color Field

General Settings

HTML Color field configuration Interface.

- **Data Type** : Select the type of field for the application. When set to HTML Color, you can select a color to be used in the form.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Set the initial value to the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Defines the amount of characters allowed for the field.
- **Validation Image** : Displays an image next to the field when it is being validated.
- **Save Variable** : Save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Defines the name for the session variable that will receive the field value.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of the field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - HTML Editor Field

General Settings

HTML Editor field configuration Interface.

- **Data Type** : Defines the type of field for the application. When set to HTML Editor, you can inform any type of character and they will be saved in HTML form.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Height** : Set the Height in pixels to the HTML Editor field.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through Javascript or PHP.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **SQL Type** : Displays the data type of field in the database.

Toolbar

HTML Editor toolbar settings Interface.

- **Properties**
  - **Position** : Location of the HTML Editor toolbar.
  - **Alignment** : Toolbar button alignment.
  - **Status** : Status Bar Display (Do not Display, Top and Bottom).
  - **Amount** : The Amount of lines of the HTML Editor toolbar.
- **Button Organization** : Positions the toolbar buttons.
- **Preview** : You can visualize the toolbar according to the previous settings.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**

- **Font** : Allows to choose the font type, that will be applied to the application field title.
- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Link

- Editor HTML

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Localization Field

General Settings

Location field Configuration Interface.

Data Type

Define the type of field. When setting it to Location, you can see a Combobox with the Languages that are part of the Project.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Reload

It links the informed text to reload the form when clicking on it.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Theme Field

General Settings

Theme field configuration Interface.

Data Type

Define the field type to Theme. it shows a combobox with the list of themes set in your project.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Reload

It Reloads the form when modifying the value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**

- **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Youtube Field

General Settings

YouTube field Configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Amount of characters

Define the number of characters allowed for the field.

Display Mode

Define the mode to display the video in the application.

Width

Video Width in pixels.

Height

Video Height in pixels.

Disabled Field

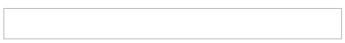
Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.



- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.

- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Google Maps Field

General Settings

Google Maps field Configuration Interface.

- **Data Type** : Select the type of field for the application. When it is defined as Google Maps, it will use the Google Maps API to display the map in the Form Applications
- **Label** : Set the title to the field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Home** : Defines what type of parameters will be used in the API.

Home Configuration Interface.

- **Display Mode** : Indicates the display mode of the map. It can be opened in a Modal or in a new Window.
- **Width** : Defines the width of the map that is going to be displayed.
- **Height** : Defines the height of the map that is going to be displayed.
- **Zoom** : Defines the initial Zoom (available from the Google API) of the Map location.
- **API Key** : API Key for authorization to use Google Maps in the Application. (Required only for the versions 2 or earlier of the Google API.)

  The API Key is a unique key, composed by a string(text) alphanumeric, which is the license to use the service. When you subscribe to use the service, the key is tied to the domain and the directory of the server. All the pages that use the API needs to be in the same directory that was used for the subscription. In case you have a web server on your local machine, you just need to possess a key for testing, and to do this you only need to set http://localhost in the domain of the subscription.

  To get your API Key access the site by clicking here

- **Link Type** : Defines how the link will be displayed.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.

- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Link 🔗

- Data entry marks

Related Videos ▷

- Form field types
- Form overview
- Calculating distances and routes with Google maps API

Form - Image (Database) Field

General Settings

Image (Database) field Configuration Interface.

Data Type

Define the type of field. When setting it to an Image (Database), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Image Border

Width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Progress bar

It displays a progress bar when sending the files to the server.

Upload area

It displays a drag and drop area to upload the file.

Upload area icon

Font-awesome icon for viewing in the file upload zone.

Clickable upload area

Hides the upload button and turns the upload area responsible for the upload function.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Display link only

It allows showing a link to open the image in another window with the original size. (with no re-dimensioning)

Open in Another Window

Allows to open the image in another window.

Extensions and upload size

This setting defines the extensions allowed for uploading and the maximum size of each one.

This setting is important for the security of your project, as it prevents unwanted files from being uploaded.

> **Note** If no value is informed, all extensions will be allowed. The file size will be limited by your PHP configuration.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Multi-upload

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.

- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]
- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.
- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.

- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.

- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Upload files to cloud
- Upload of Multiple Images

Related Videos ▷

- Form field types
- Form overview
- Upload and multi upload fields
- Working with images 1/2
- Working with images 2/2
- Saving uploaded files into Google Drive
- Macro sc_field_init_off
- Upload field with customizable rules (size and format)

Form - Image (File Name) Field

General Settings

Data Type

Define the type of field. When setting it to Image (File Name), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Image Border

The width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Progress bar

It displays a progress bar when sending the files to the server.

Upload area

It displays a drag and drop area to upload the file.

Upload area icon

Font-awesome icon for viewing in the file upload zone.

Clickable upload area

Hides the upload button and turns the upload area responsible for the upload function.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API

Keep file after uploading via API

It also saves the file in the default folder for images or documents after uploading to a cloud storage API.

Deletion in the storage API

When deleted in the application, the file will also be removed from the folder in the cloud storage API.

Increment file

It inserts a number increment into the file name if there is already another one with the same name in the upload folder.

Subdirectory for local storage

It sets the Sub-folder name of the stored files. It is relative to the directory of Document upload (see the Settings). It is possible to use global variables or local variables to format the name of the sub-folder.

Create Subfolder

It creates the sub-folder if not already created.

Image Caching

Time in minutes to store the image in the cache.

Hide image name

It displays only the image, without the name.

Files Deletion

It Deletes the files from the directory when deleting a record from the database.

Display link only

It allows showing a link to open the image in another window with the original size. (with no re-dimensioning)

Open in Another Window

Allows to open the image in another window.

File Size

It allows defining the field to store the file size in the database.

Extensions and upload size

This setting defines the extensions allowed for uploading and the maximum size of each one.

This setting is important for the security of your project, as it prevents unwanted files from being uploaded.

> **Note** If no value is informed, all extensions will be allowed. The file size will be limited by your PHP configuration.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Multi-upload

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.
- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]
- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.

- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Upload files to cloud
- Upload of Multiple Images

Related Videos ▷

- Form field types
- Form overview
- Upload and multi upload fields
- Working with images 1/2
- Working with images 2/2
- Saving uploaded files into Google Drive
- Macro sc_field_init_off
- Upload field with customizable rules (size and format)

Form - Document (Database) Field

General Settings

Document (Database) field Configuration Interface.

Data Type

Define the type of field. When setting it to Document (Database), all the document files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Icon

It displays an icon beside the field to identify the document type.

Progress bar

It displays a progress bar when sending the files to the server.

Upload area

It displays a drag and drop area to upload the file.

Upload area icon

Font-awesome icon for viewing in the file upload zone.

Clickable upload area

Hides the upload button and turns the upload area responsible for the upload function.

File Name

It allows defining the field to store the file name in the database.

File Size

It allows defining the field to store the file size in the database.

Extensions and upload size

This setting defines the extensions allowed for uploading and the maximum size of each one.

This setting is important for the security of your project, as it prevents unwanted files from being uploaded.

> **Note** If no value is informed, all extensions will be allowed. The file size will be limited by your PHP configuration.

Allow download in read only fields

Allow file download even when field is read-only.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Multi-upload

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.
- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]

- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.
- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.

- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Upload files to cloud
- Upload of Multiple Images

Related Videos ▷

- Form field types
- Form overview
- Upload and multi upload fields
- Working with images 1/2
- Working with images 2/2
- Saving uploaded files into Google Drive
- Macro sc_field_init_off
- Upload field with customizable rules (size and format)

Form - Document (File Name) Field

General Settings

Document (File Name) field Configuration Interface.

Data Type

Define the type of field. When setting it to Image (File Name), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Progress bar

It displays a progress bar when sending the files to the server.

Upload area

It displays a drag and drop area to upload the file.

Upload area icon

Font-awesome icon for viewing in the file upload zone.

Clickable upload area

Hides the upload button and turns the upload area responsible for the upload function.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API

Keep file after uploading via API

It also saves the file in the default folder for images or documents after uploading to a cloud storage API.

Deletion in the storage API

When deleted in the application, the file will also be removed from the folder in the cloud storage API.

Increment file

It inserts a number increment into the file name if there is already another one with the same name in the upload folder.

Subdirectory for local storage

It sets the Sub-folder name of the stored files. It is relative to the directory of Document upload (see the Settings). It is possible to use global variables or local variables to format the name of the sub-folder.

Create Subfolder

It creates the sub-folder if not already created.

Icon

It displays an icon beside the field to identify the document type.

File Size

It allows defining the field to store the file size in the database.

Extensions and upload size

This setting defines the extensions allowed for uploading and the maximum size of each one.

This setting is important for the security of your project, as it prevents unwanted files from being uploaded.

> **Note** If no value is informed, all extensions will be allowed. The file size will be limited by your PHP configuration.

Allow download in read only fields

When enabled, this option allows file download even when the field is set to read-only.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

Files Deletion

It Deletes the files from the directory when deleting a record from the database.

SQL Type

It informs the data type of field in the database.

Multi-upload

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

 Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

 Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.
- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]
- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.
- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position

(left,right,center and justify).
    - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the title of the field.
    - **Height** : To define a height for the title of the field.
- **CSS of the Field**
    - **Font** : Allows to choose the font type, that will be applied to the application field.
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the field.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
    - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the field.
    - **Height** : To define a height for the field.
- **CSS of the Input Object**
    - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the Input Object.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
    - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the Input Object.
    - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Upload files to cloud
- Upload of Multiple Images

Related Videos ▷

- Form field types
- Form overview
- Upload and multi upload fields
- Working with images 1/2
- Working with images 2/2
- Saving uploaded files into Google Drive
- Macro sc_field_init_off
- Upload field with customizable rules (size and format)

Form - Barcode Field

General Settings

Configuration Interface of the Barcode Field.

- **Data Type** : DataType of the field for the application.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Disabled Field** : Defines if the field will be disabled in the “Insert Mode” , “Update Mode” or in “Insert and Update Mode”.
- **SQL Type** : Database field type.

Values Format

Configuration Interface of the Barcode Field.

- **Type** : Type of Barcode.
- **Text** : Barcode Text for illustration purposes.
- There are **18 types of barcodes**, that are listed below:

Barcode configuration interface.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Link

- Data entry marks

Related Videos

- How to create and use barcodes
- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Select Field

General Settings

Select field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Select, you can select multiple option from a combo box (Select Field).
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.
- **Use Select2** : Uses the new component for data selection, allowing searches within the select.
- **Display Select2 search area** : Sets whether to display the search field within Select2.

Initial Value Configuration Interface.

- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the field size in bytes. It is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through Javascript or PHP.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

 Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

  Selecting the lookup type.

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimitation.

      - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Height** : Defines the height for the select object.
    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

      - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
      - **Title** : Text displayed on the title of the Select object.
    - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
    - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

      - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
    - **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.
    - **Link** : Allows to create a link to another form allowing to manipulate the list displayed on the select field. After the manipulation, the select object it updated automatically.

  - **Lookup Method - Manual**

    This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table

that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Height** : Defines the height for the select object.
  - **Multiple Values (delimiter)**

    You can store various values for the select field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Height** : Defines the height for the select object.
  - **Multiple Values (position)**

    Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

    For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

    The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

    Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

    Setting up Multiple Values (position)

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
    - **Start** : Starting position of the string that is going to be stored. The first position is always 1.

- **Size** : Amount of bytes that is going to occupy in the string.
- **Height** : Defines the height for the select object.

- **Multiple Values (binary)**

  Allows to store a diversity of information selected on the field in only one value.

  In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  To store the data, it is done an internal sum of the decimal number. See the examples below.

  Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 2 (Culture) = **3**

  Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

  Editing Lookup Configuration Multiple Values (Binary)

  - **Label** : Text that will be displayed in the item list of the select field.
  - **Height** : Defines the height for the select object.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.

- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  Displaying the original lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Lookup Method - Actual value

This lookup is used to list all the values in the selected field.

This lookup will apply a “distinct” to your SQL query.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio,

Select, Text, etc
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Multi select Combobox
- Many fields (double select field)
- Many to many (checkbox field)
- Data validation
- Macro sc_focus_edit
- Macro sc_field_init_off

Related Videos

Form - Double Select Field

General Settings

Double Select field configuration Interface.

- **Data Type** : Select the type of field for the application. When set to Double Select you can have multiple options selected.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Defines the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to set the max amount of characters to be typed in.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through Javascript or PHP.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Displays the data type of field in the database.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

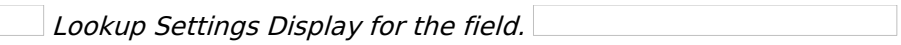
- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, getting these values from the database.

Lookup Settings Display for the field. 
Automatic Lookup Interface..

- **SQL Select Statement** : Defines the SQL command that will get the values displayed on the form field. The command needs to have the following structure:

  SELECT key_field , display_field FROM table

  The value of the key_field will be stored in the table field.

- **Height** : Set the height(lines) of the field interface.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter than (;).
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top,

text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the field.
    - **Height** : To define a height for the field.
- **CSS of the Input Object**
    - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the Input Object.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
    - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the Input Object.
    - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Combobox
- Combobox with data edition
- Multi select Combobox
- Many-field types (double select field)
- Many-to-many (checkbox field)
- Data validation
- Macro sc_focus_edit
- Macro sc_field_init_off

Related Videos

Form - Checkbox Field

General Settings

CheckBox field configuration Interface.

- **Data Type** : Select the type of field for the application. When set to CheckBox, you can have multiple options selected.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Defines the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to determine the max amount of characters to be typed in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through JavaScript or PHP. .
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

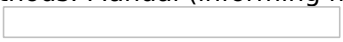
- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).  *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the CheckBox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
    - **Option check all** : Displays two options on the field to check and uncheck all.

    - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

      - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
    - **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

  - **Lookup Method - Manual**

    This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

    Manual Lookup Interface.

    - **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).

      - **Single Value** :

        Setting up Single Value Lookup

        - **Label** : Text that will be displayed in the item list of the checkBox.
        - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
        - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
        - **Columns** : Set the amount of columns, for the list of items.

      - **Multiple Values (delimiter)**

        You can store various values for the checkBox field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

Setting up Multiple Values (delimiter)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Columns** : Set amount of columns, for the list of items.

- **Multiple Values (position)**

  Stores a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

  For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

  The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

  Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

  Setting up Multiple Values (position)

  - **Label** : Text that will be displayed in the item list of the checkBox.
  - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
  - **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
  - **Start** : Starting position of the string that is going to be stored. The first position is always 1.
  - **Size** : Amount of bytes that is going to occupy in the string.
  - **Columns** : Set the amount of columns, for the list of items.

- **Multiple Values (binary)**

  Stores a diversity of information selected on the field in only one value.

  In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  To store the data, it is done an internal sum of the decimal number. See the examples below.

  Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

Setting up Multiple Values (Binary)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Columns** : Allows you to inform the amount of columns, for the list of items.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Saves all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Refreshes the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Option check all** : Displays two options on the field to check and uncheck all.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Autocomplete
- Combobox with data edition
- Multi select Combobox
- Many-to-many (double select field)
- Many-to-many (checkbox field)
- Data validation
- Macro sc_focus_edit
- Macro sc_field_init_off

Related Videos

- Form field types
- Form overview

Form - Radio Field

General Settings

Radio field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Radio, your allowed to select one of the options listed.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to determine the max amount of characters to be typed in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through Javascript or PHP. .
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

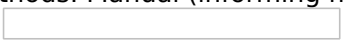
Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example,

the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).  *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Columns** : Allows you to inform the amount of columns, for the list of items.

    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

      - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
      - **Title** : Text displayed on the title of the Select object.
    - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
    - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

      - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
    - **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

  - **Lookup Method - Manual**

    This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

    Manual Lookup Interface.

    - **Label** : Text that will be displayed in the item list of the radio.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Insert Button** : Adds to the list the values informed on the fields Label and Value.
    - **Update Button** : Modifies the attributes of the selected item.
    - **Remove Button** : Remove the selected item from the list.
    - **Clear Button** : Clear the Fields.
    - **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on load lookup definition.
    - **Load lookup definitions** : Allows to refresh the list of values predefined when using the

lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  Displaying the original lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Settings

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Multi select Combobox
- Many fields (double select field)
- Many to many (checkbox field)
- Data validation
- Macro sc_focus_edit
- Macro sc_field_init_off

Related Videos

Form - Text Auto-Complete Field

General Settings

Text auto complete field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Text auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal Text for the data.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You’ll inform the Defined Value here.
- **Use Select2**: Uses the new component for data selection, allowing searches within the select.
- **Amount of characters for the Select2** : Sets the number of characters to start the search in Select2.
- **Amount of lines for the Select2** : Sets the maximum number of rows to list the search result in Select2.
- **Width for the Select2** : Sets a width for the area for the Select2.

Initial Value Configuration Interface.

- **Amount of characters** : Define the amount of characters allowed for the field.
- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Complete to the Left** : Allows to define the character that will be used to complete the value to the left that the user typed in to the max size of the field defined in the Field size in database option.
- **Field size in database** : Determines the field size in bytes. it is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Validate on submit** : Validate the field only when the form is submitted.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

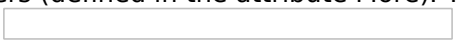
Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed. 

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.

- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below. *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc

- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed

- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Autocomplete
- Combobox with data edition
- Multi select Combobox
- Many fields (double select field)
- Many to many (checkbox field)
- Data validation
- Macro sc_focus_edit
- Macro sc_field_init_off

Related Videos

- Form field types
- Form overview

Form - Number Auto-Complete Field

General Settings

Number auto complete field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Number auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal number for the data.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Define the amount of characters allowed for the field.
- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Complete to the Left** : Allows to define the character that will be used to complete the value to the left that the user typed in to the max size of the field defined in the Field size in database option.
- **Field size in database** : Determines the field size in bytes. It is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Validate on submit** : Validate the field only when the form is submitted.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed.

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example

below. *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.

- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse

leaves the help icon.

- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Data entry marks
- Autocomplete
- Combobox with data edition
- Multi select Combobox
- Many-to-many (double select field)
- Form field types
- Many-to-many (checkbox field)
- Form overview
- Data validation
- Macro sc_focus_edit
- Macro sc_field_init_off

Related Videos ▷

Form - Signature Field

General Settings

Configuration Interface of the Signature Field.

The signature field will help you creating more sofisticated forms and making it possible to store signatures in your database. Inside our development environment we have specific settings that will help you to customize your field, those options are:

- **Data Type** : You can define the type of field for the application. When it is defined as a text, it accepts letters, numbers and special characters.
- **Label** : Lets you define a label to the field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Background color** : Defines a color to the field background by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Pen color**: Set a color to the pen by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Width** : Lets you define a width to the field.
- **Height** : Set a height to the field.
- **Subtitle** : Defines the subtitle that will be displayed beside the field.
- **Initial Value** : Lets you define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
- **Save Variable** : Save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Disabled Field** : Define if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : Displays the HTML Object that will be used to display the field in the form.
- **SQL Type** : Displays the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of ScriptCase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **CSS of the Title**
  - **Font** : Select the font type, that will be applied to the application field title.
  - **Font Size** : Defines the font size, that will be applied to the application field title.
  - **Font Color** : Choose a color to the font by using a valid hexadecimal color value or from the color picker.
  - **Background Color** : You can define the color for the field by using a valid hexadecimal color value or from the color picker.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Set the underline style to the font.
  - **Border style** : Choose a style for the border.
  - **Collapse** : Defines the collapse for the border.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Choose a color for the border, using a color palette to apply to the title.
  - **Horizontal Alignment** : Position the label of the field in the wanted position (left,right,center and justify).
  - **Vertical Alignment** : Position the label of the field in the wanted position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : Defines a width for the title of the field.
  - **Height** : Set a height for the title of the field.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Link

- Digital Signature

Related Videos ▷

- Form field types
- Form overview

Form - Rating Field

General Settings

This field allows the developer to create a field for rating using stars (or any other image), where the final user can select a rate option.

Attribute descriptions

Data Type

Define the field type to **Rating**.

Label

The Label option lets you define the title of a field. Example: If the database field name is “**Stars**”, You can display a different name for the user, like “**Stars**”.

Besides use a fixed text, the **Label** attribute allows the use of **Langs** to define the field title, allowing the internationalization of your application.

Label below field

Defines the message to be displayed below the field.

Subtitle

Define the subtitle of the field, below the ratings. **Example**: “Thank you for your feedback!”.

As in the **Label**, the **subtitle** attribute also allows the use of **Langs** for the internationalization of your application.

Amount of icons

It defines how many icons it will display in the field. The value set in this attribute must be according to the evaluation rules.

Initial Value (type)

Allow the definition of an initial value for the field when the form application is in insert mode. The only option available in this field is **Defined**. Selecting this option the attribute **Initial Value** will be showed for set the value.

Example: The initial value is 3 when inserting a new record, the rate field will initiate with 3 stars already selected.

> The value defined in this attribute will overlap any value defined previously.

Save variable

It allows saving the value of the field in a session variable(Global variable) to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the **session variable** that will receive the field value.

We must inform only the variable name, - **var_rating**.

We can get the value as a global variable.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of ScriptCase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **CSS of the Title**
  - **Font** : Select the font type, that will be applied to the application field title.
  - **Font Size** : Defines the font size, that will be applied to the application field title.
  - **Font Color** : Choose a color to the font by using a valid hexadecimal color value or from the color picker.
  - **Background Color** : You can define the color for the field by using a valid hexadecimal color value or from the color picker.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Set the underline style to the font.
  - **Border style** : Choose a style for the border.
  - **Collapse** : Defines the collapse for the border.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Choose a color for the border, using a color palette to apply to the title.
  - **Horizontal Alignment** : Position the label of the field in the wanted position (left,right,center and justify).
  - **Vertical Alignment** : Position the label of the field in the wanted position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : Defines a width for the title of the field.
  - **Height** : Set a height for the title of the field.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Data entry marks
- Validation

Related Videos

- Form field types
- Form overview
- Macro sc_field_display
- Macro sc_field_disabled
- Macro sc_set_edit
- Macro sc_field_readonly
- Macro sc_field_disabled_record

Form - Rating Smile Field

General settings

This field type allows the final user evaluate with an emoji that goes from **Very Bad** to **Excellent**.

Data type

Defines the data type of the field in the application. In this case we should select **Rating Smile**.

Label

Defines the title that will be shown in the field when the application is running. The used terms in the interface are essential to the system has good usability, we should use comum name and terms for the final user, instead of the default values of the system.

For example, for the rating smile field, the client would have a better understanding of the field function when the label is **Product evaluation**.

Besides a fixed text, the **Label** allow to use of lang to define the field title, this way is possible to use the application internationalization.

Label below field

Allows you to define a message that will be displayed under the field, which can serve as a help or caption for example.

Subtitle

Define a subtitle that will be shown beside the field. Ex.: Thank you for the feedback!

Below is one example of an application subtitle:

Values and hints

Allow to the developer to define the values to each emoji taht will be saved at the data base and allow to define a hint **(It is show passing the mouse above the emoji)**

Display format

Allows the developer to define the format that the emojis will be shown.

In case the first option is defined, the emojis will be shown like this:

In case the second option is defined, the emojis will be shown like this:

Icon size

Allows defining the icon size in pixels.

Rating padding

Allows for defining the padding value between the emojis.

Icon colors

Allows the icon colors to be defined in each emoji.

Initial value (type)

Allows defining a default value to the field when the application will be in insert mode. Is possible to choose between two options:

Default value: Selecting this option, the default value will be available, where we should input the field's default value. For example, my default value is **Allan**, inserting a new register, the field **Seller Name** will start with Allan.

Save variable

It allows saving a session variable (global variable) with the field value, to be used in other applications.

For example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable name

In this one we should define the variable's name in session, available at the last item, which will receive the field value.

We should inform only the variable's name, **var_seller**.

The recovery of the value is done in the form of a global variable.

Disable field

Allow to Disable field, impossible to change the defined value.

Available options are:
No - Default value, this option don't disable the field.
Update mode - This option disable the field only at update. **Insert mode** -This option disable the field only at insert. **Insert/Update mode** -This option disable the field in both insert and update.

HTML type

HTML object used to show the form field.

SQL type

Infomrs the field type at the data base.

Display Sttings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Form - Rating Thumbs Field

General settings

This field type allows the final user to save data with the icons like or dislike.

Data type

Defines the data type of the field in the application. In this case we should select **Rating Thumbs**.

Label

Defines the title that will be shown in the field when the application is running. The used terms in the interface are essential to the system has good usability, we should use comum name and terms for the final user, instead of the default values of the system.

For example, for the rating thumbs field, the client would have a better understanding of the field function when the label is **Product evaluation**.

Besides a fixed text, the **Label** allow to use of lang to define the field title, this way is possible to use the application internationalization.

Label below field

Allows you to define a message that will be displayed under the field, which can serve as a help or caption for example.

Subtitle

Define a subtitle that will be shown beside the field. Ex.: Welcome to help us with your feedback!

Below is one example of an application subtitle:

Values and hints

Allow to the developer to define the values to each icon taht will be saved at the database and allow to define a hint **(It is shown passing the mouse above the icon)**

Display format

Allows the developer to define the format that the emojis will be shown.

In case the first option is defined, the emojis will be shown like this:

In case the second option is defined, the emojis will be shown like this:

Icon size

Allows defining the icon size in pixels.

Rating padding

Allows for defining the padding value between the emojis.

Icon colors

Allows the icon colors to be defined in each emoji.

Initial value (type)

Allows defining a default value to the field when the application will be in insert mode. Is possible to choose between two options:

Default value: Selecting this option, the default value will be available, where we should input the field's default value. For example, my default value is **Allan**, inserting a new register, the field **Seller Name** will start with Allan.

Save variable

It allows saving a session variable (global variable) with the field value, to be used in other applications.

For example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable name

In this one we should define the variable's name in session, available at the last item, which will receive the field value.

We should inform only the variable's name, **var_seller**.

The recovery of the value is done in the form of a global variable.

Disable field

Allow to Disable field, impossible to change the defined value.

Available options are:
No - Default value, this option don't disable the field.
Update mode - This option disable the field only at update. **Insert mode** -This option disable the field only at insert. **Insert/Update mode** -This option disable the field in both insert and update.

HTML type

HTML object used to show the form field.

SQL type

Infomrs the field type at the data base.

Display Sttings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

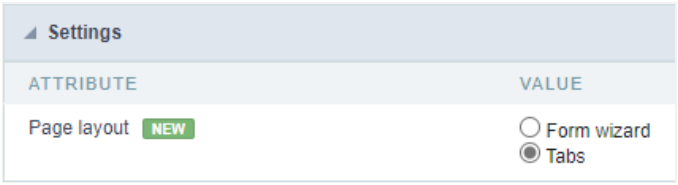
Form Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

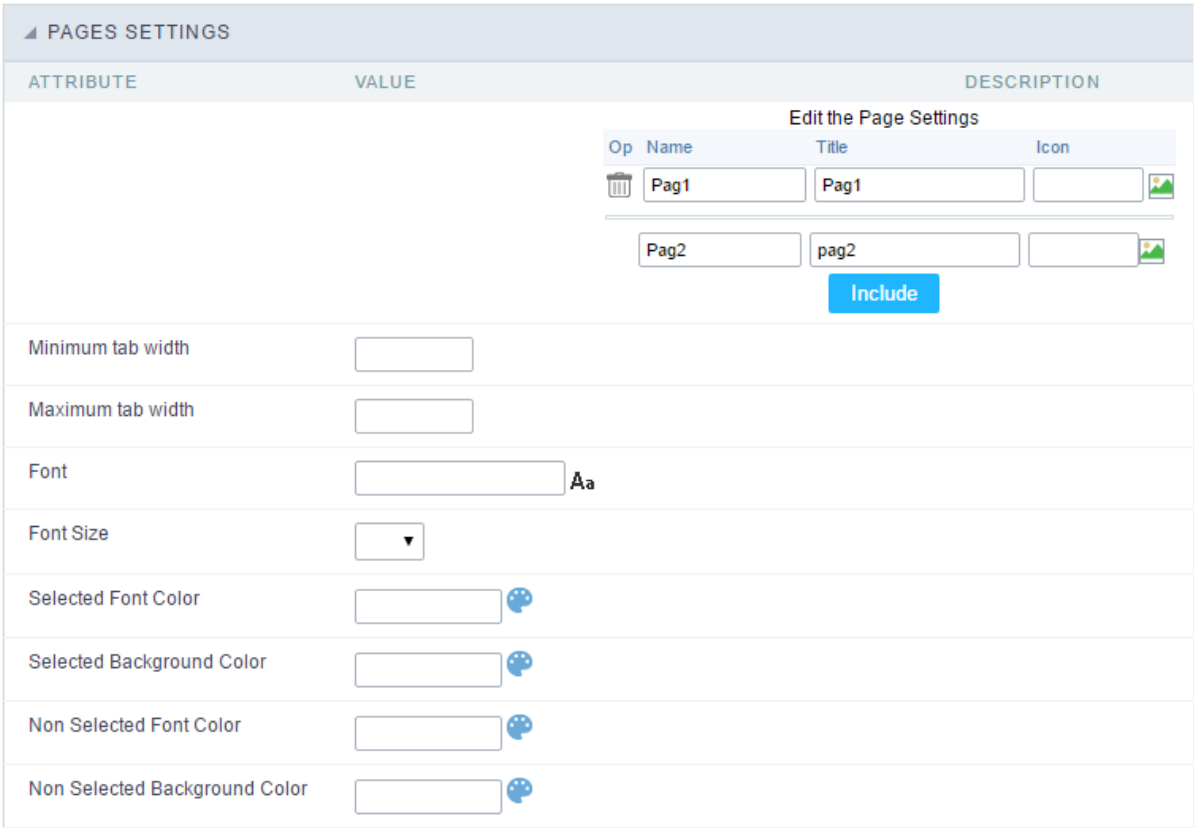
Pages

A page is a container of blocks. Each application has at least one page by default. In Form, Control, and Search applications, there is possible to create many pages. The pages are like Tabs, where each tab contains one or more blocks.

The pages can be shown as **Tabs** or in form of a **Form Wizard**. In the option below you can select the type of your Page Layout.

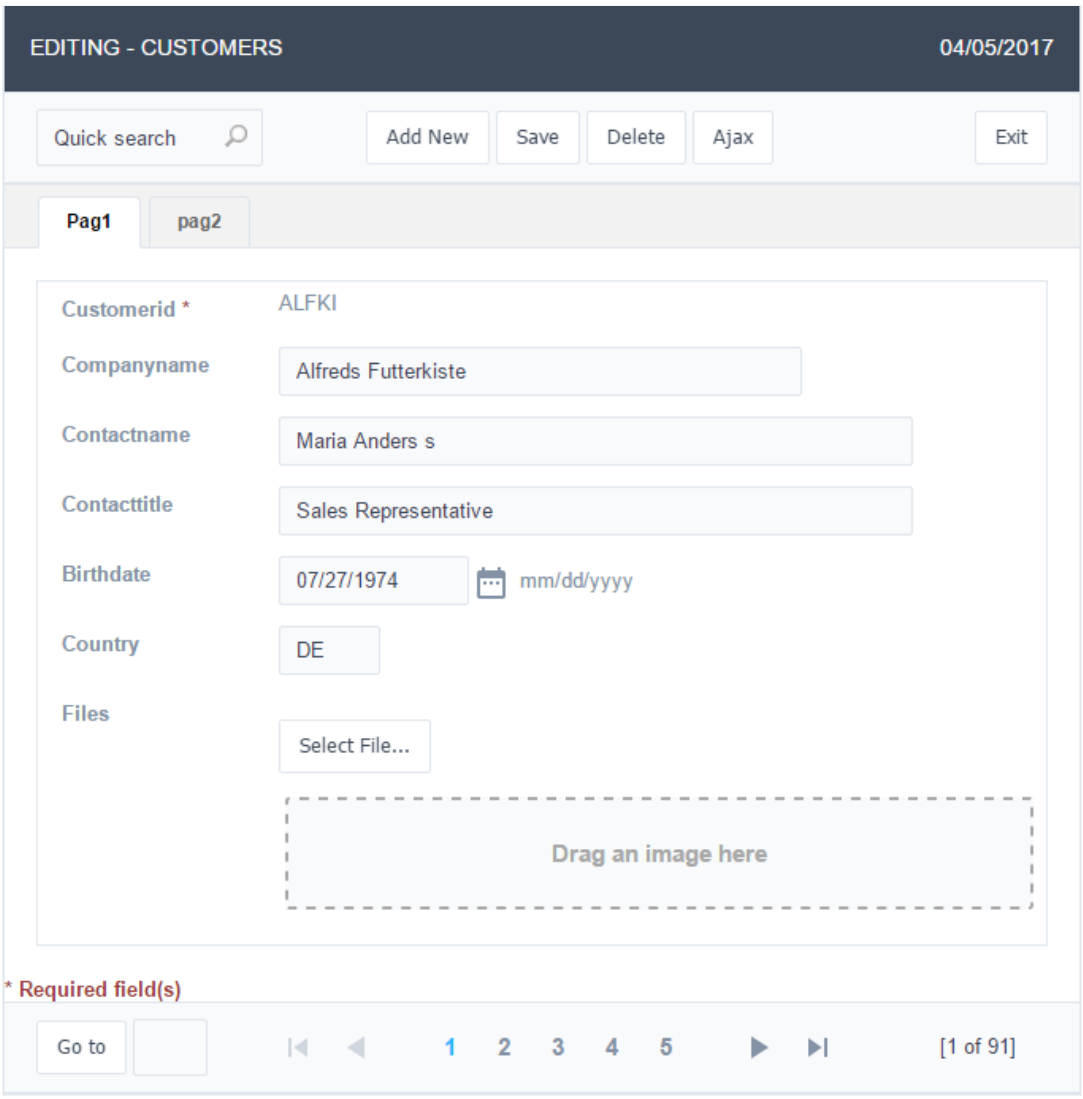


Pages as Tab



Pages (available only in the Form, Control and Search applications) configuration Interface.

See the example below of the Form application using two Pages: General Data and Documents.

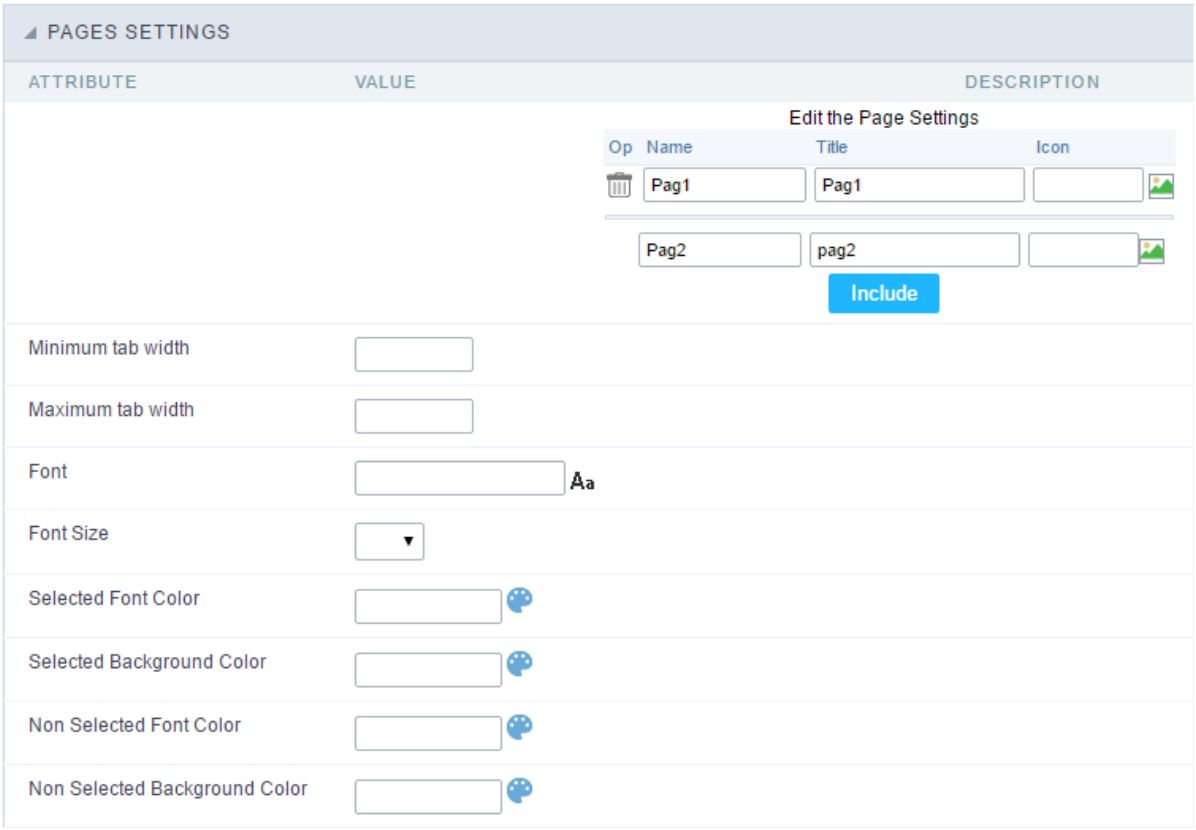


Form Application using Pages

feature.

Pages Settings

The form application already comes with a default page, identified as “Pag1”, but it is possible to rename it. Use the pages when you have an application that contains many fields. A form with more than 20 fields in a vertical way is challenging to use. So you could arrange the fields into the blocks and the blocks into the pages.



Pages Configuration Interface.

Including a new page,

To include a new page, enter the information about the name and the label of the new page, select an image if you wish, then click on the button “Include”.

Deleting a page

To delete a page, click on the Trash icon corresponding to the page line.

Common Settings

Font

Set the font-family of the pages titles. By clicking on the right side icon, you can select the font-family from a list.

Font Size

Set the font size of the pages titles.

Selected Font Color

Set the font color of the selected page.

Selected Background Color

Set the Background Color of the selected page.

Non Selected Font Color

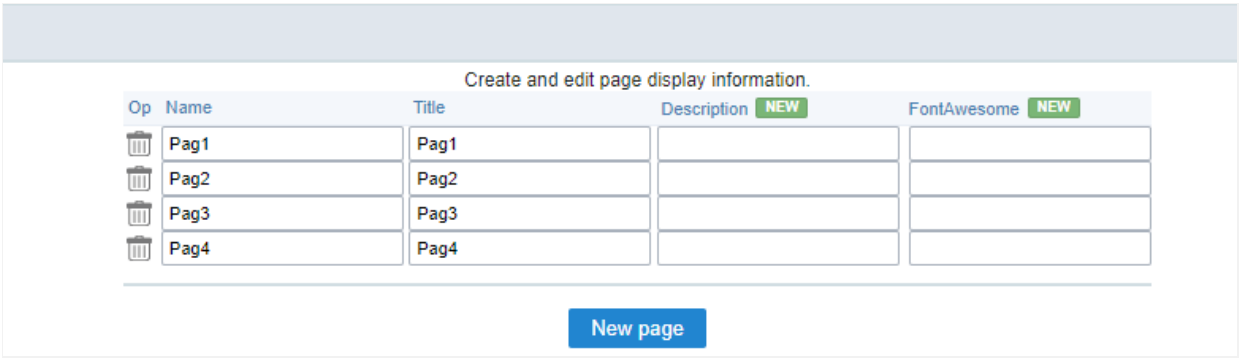
Set the font color of the non selected pages.

Non Selected Background Color

Set the Background Color of the non selected pages.

Pages as Wizard

Step Configuration



Op This option is for delete the page you want.

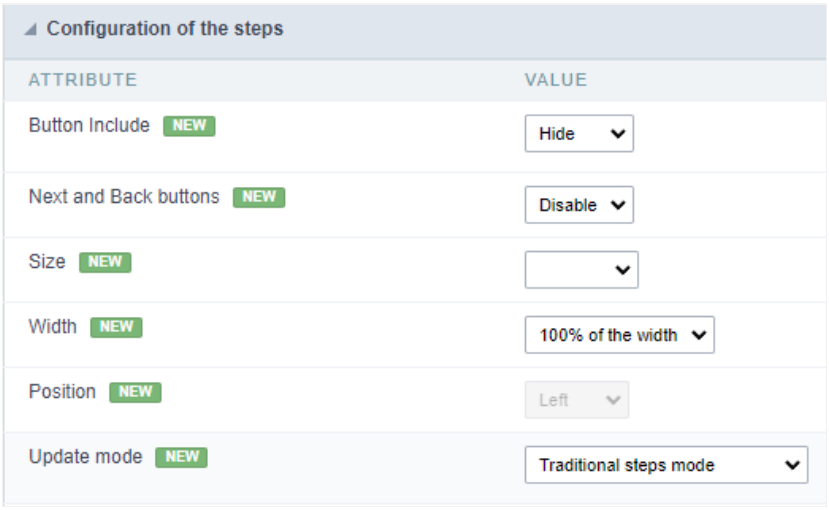
Name
Its the name of the page.

Title In this option the user can give a title for the page.

Description In this option the user can configure a description. The description will be shown on the step of your Steps.

Display In this option you can select the icon that will be shown above the Step number, you can use Icon or Font Awesome. By default there is none icon enabled, the user need to select one to be shown.

Pages Steps Format



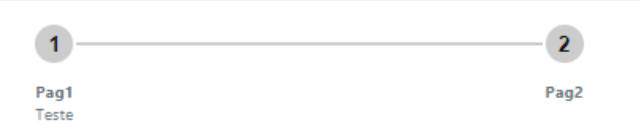
Button Include
This option allow the user to configure if the Insert Button will be shown on all steps, but will be disabled, or if want to hide the button to be shown only on the last step.

Next and Back Buttons This option allow the user to configure if the Back Button will be shown on all steps, but will be disabled, or if want to hide the button to be shown only after you advance the form.

Size This option defines the size of the steps visualization, you can chose between Samll, Medium or Large.

Example:

Width
This option define the width of the steps, you can chose between Minimun or 100% of the width.
Example with 100% of the width:



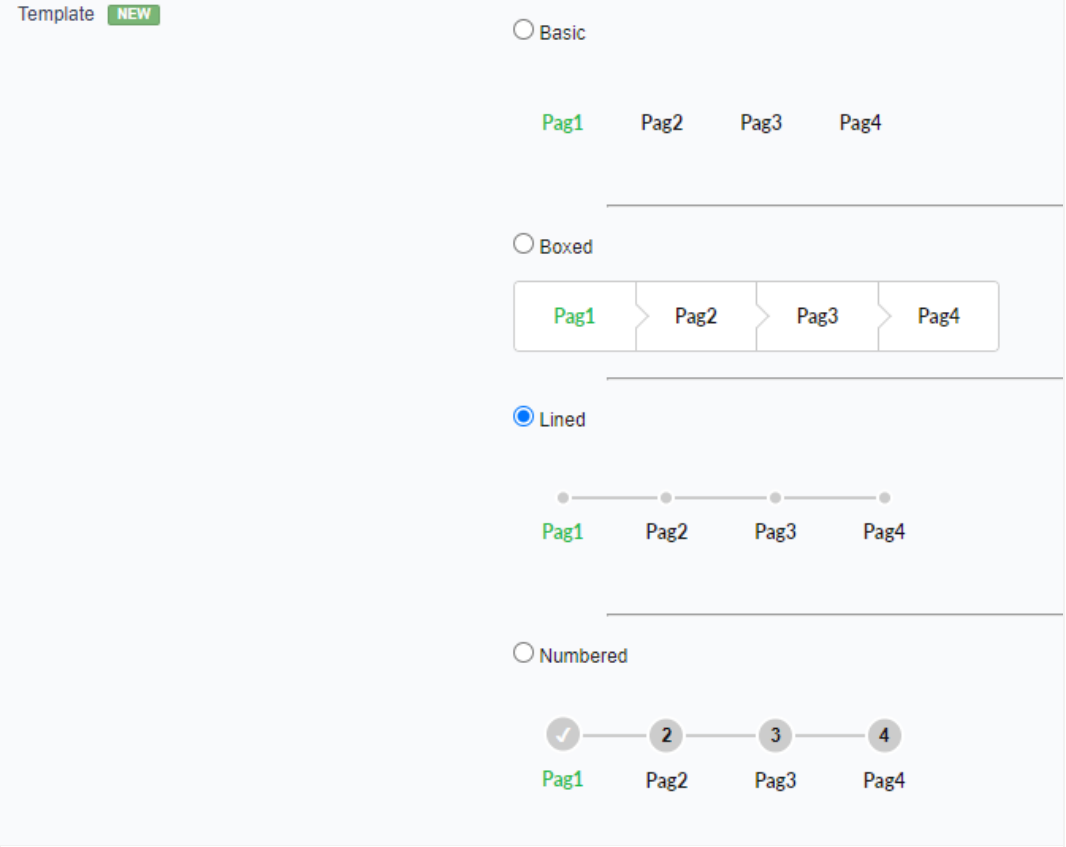
Position This option defines where the steps will be located, you can chose between Left, Center or Right.

Update Mode
Defines how the form Steps will be used when updating records:

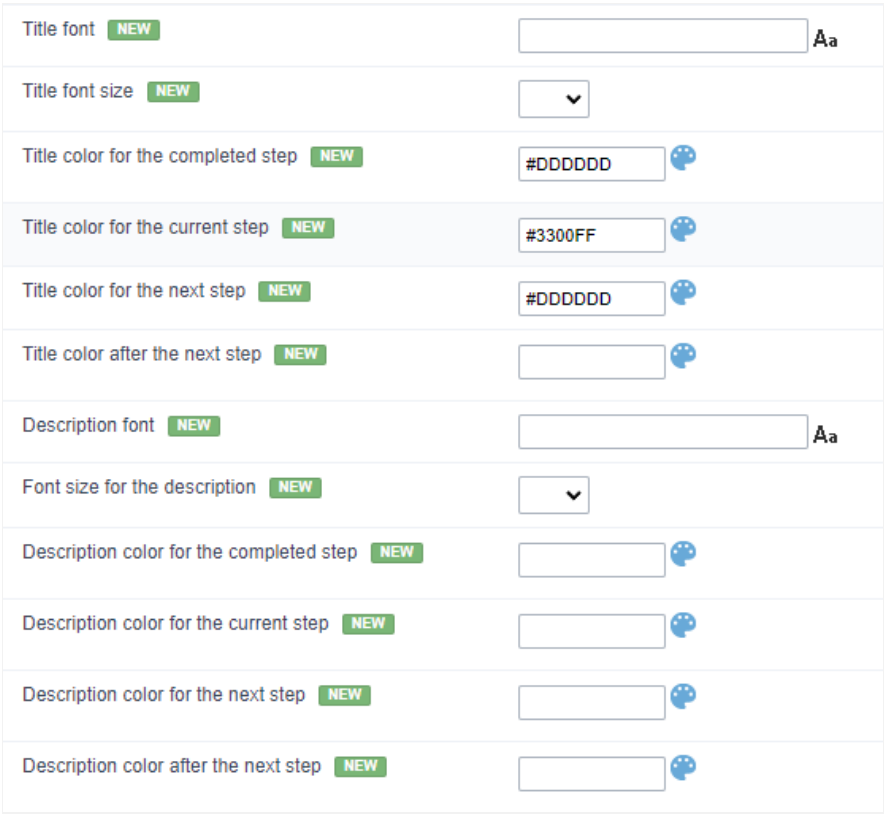
- **Traditional Steps Mode**: Traditional Steps mode (every step is mandatory).
- **Steps with navigation enabled**: The Steps interface will be used, but with no button to next or back from the Steps, being able to navigate any step freely.
- **Pages Interface**: The traditional page interface will be used.

Template

This option the user is can select how the Steps will look like. In the image there is a preview of the option selected.



Pages Steps Format



Title Font Defines a font to be used in step titles. If not informed, it will use the same font defined in the application theme.

Title Font Size Defines the font size for step titles. If not informed, the same font size defined in the application theme will be applied.

Title color for the completed steps Defines the font color for the title of steps already completed. If not informed, the same color defined in the application theme will be applied.

Title color for the current step Sets the font color for the title of the current step. If not informed, the same color defined in the application theme will be applied.

Title color for the next step Defines the font color for the next step title. If not informed, the same color defined in the application theme will be applied.

The color after next step Defines the font color for the title of steps after the next step. If not informed, the same color defined in the application theme will be applied.

Description Font Defines the font for the step description. If not informed, the same font defined in the application theme will be applied.

Font size for description Defines the font size for step descriptions. If not informed, the same font size defined in the application theme will be applied.

Description color for the completed step Defines the font color for the description of steps already completed. If not informed, the same color defined in the application theme will be applied.

Description color for the current step Defines the font color for the current step description. If not informed, the same font color defined in the application theme will be applied.

Description color for the next step Defines the font color for the next step description. If not informed, the same font color defined in the application theme will be applied.

Description color after the next step Defines the font color for the step description after the next step. If not informed, the same font color defined in the application theme will be applied.

Layout

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Blocks

Blocks are “containers” where you can position the application fieldSlides of Forms, Controls, or Grids.

Scriptcase creates applications with one block by default. You can add more blocks as you wish, to organize it in the best way.

See below, the Columns Organization, and where you can define the position of the next block: beside or below the current one.

Application Block configuration

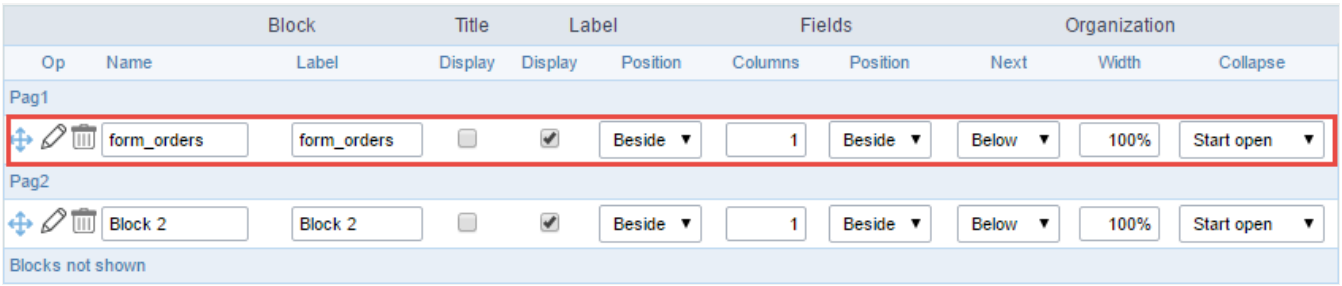
On the left side of each block, there are two icons, the first one to edit the information of the block and the second one

to delete the block.

Organizing the position of the Blocks

See below how to modify the display order of the Blocks in one Page.

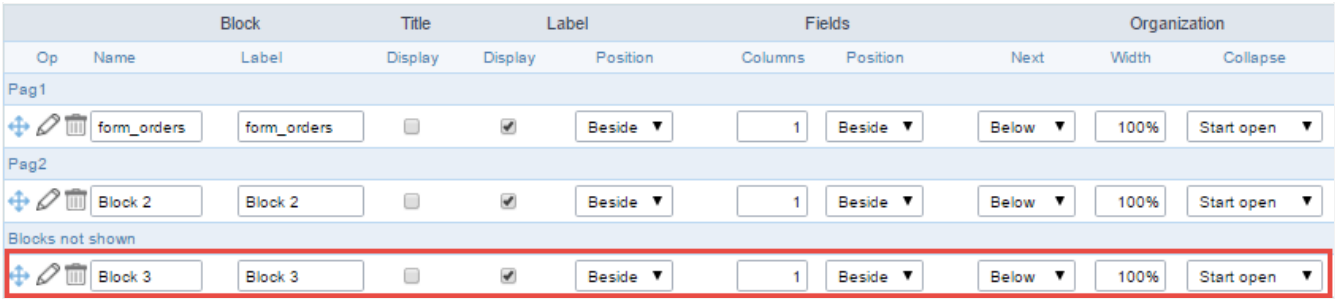
Click and drag the block that you desire to modify to its new position.



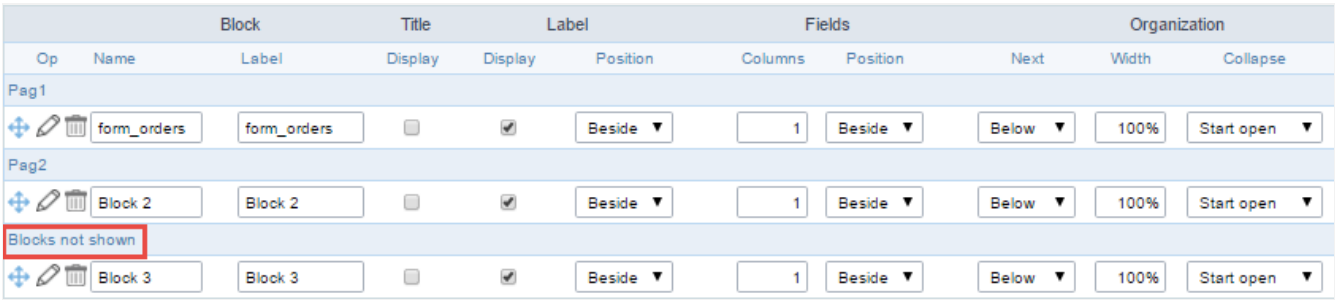
Application Block Display configuration

- See how to remove a block from the display

Click on the block desired and drag it to the item “Blocks not Shown”. This way, you can also drag the block to another page if desired. See the images below.



Application Block Display configuration



Application Block Display configuration

Block

- **Name**: The name that identifier the Block.
- **Label**: Title of the block to display in the application.

Title

- **Display**: It controls the display of the block title.

Label

- **Display**: It controls the display of the field labels of the block.
- **Position**: Options to display label :
    1. **Above**: Display the label above the field.
    2. **Beside**: Display the label beside the field.
    3. **Below**: Display the label below the field.

Fields

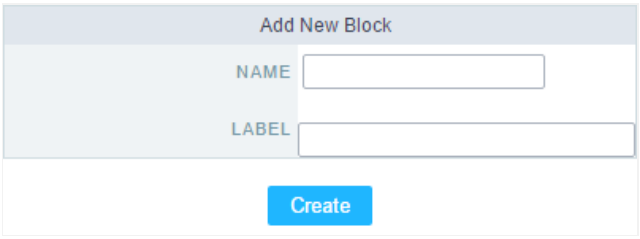
- **Columns**: Amount of columns side by side in the block.
- **Position**: The way to display the fields in the block:
    1. **Below**: Display the fields one below the other respecting the number of columns.
    2. **Beside**: Display the fields one beside the other respecting the number of columns.
    3. **Line**: Display the fields one beside the other with no tabulation.

Organization

- **Next**: The way to display the blocks in the page:
    1. **Below**: Set to show the following block below the current one.
    2. **Beside**: Set to show the following block beside the current one.
    3. **Tabs**: Set to show the following block in a different tab then the current one.
- **Width**: Set the block width in pixels or percentages. Use the symbol “%” to indicates the value in percentage.
- **Collapse**: Enables the option to close the block.

Create a New Block

To include new blocks in an Application, click on the button Create New Block. Then, enter the name and label of the block in the following interface and finish by click on Create.



Creating application blocks configuration

Name

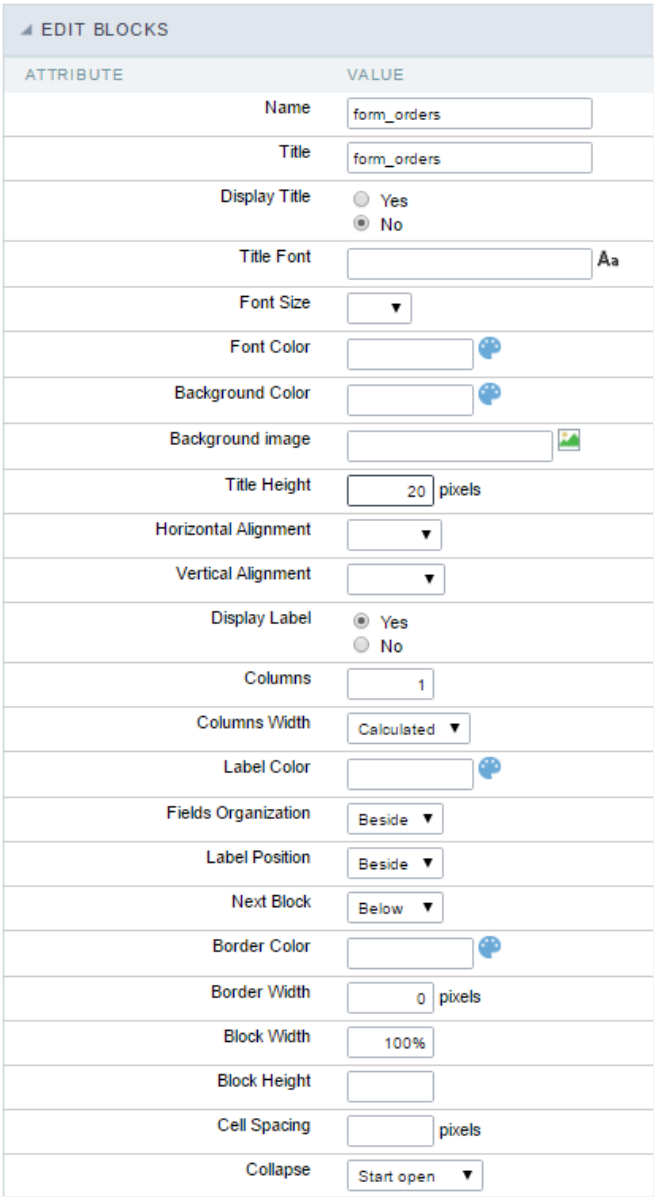
Name of the Block.

Label

Title of the block to display in the application.

Edit Blocks

To edit a block, click on the icon , that is on the left side of the block. Then you can see the following interface to define the parameters of the blocks. Click on Save to finish.



Application Block editing interface

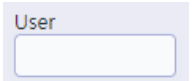
Name

Name of the block. #### Title
Block title for display. #### Display Title
This option, when active, allows displaying the block title. #### Title Font
Set the font family of the block title. #### Font Size
Set the font size of the block title. #### Font Color
Set the font color of the block title. #### Background Color
Set the Background Color of the block title. #### Background image
Set a Background image for the block title. #### Title Height
Height in pixels of the block title line. #### Horizontal Alignment
Horizontal Alignment of the block title (Left, Center, and Right). #### Vertical Alignment
Vertical Alignment of the block title (Top, Middle, and Bottom). #### Display Label
Display the labels of the fields in the block. #### Columns
Amount of field columns in a block. #### Columns Width
Set the field column width of the block. #### Label Color
Color of the field labels. #### Fields Organization
The way to display the fields in the block. #### Label Position
Set the position of the field labels of the block.

The options are:

- **Beside** - This option positions the label on the right side of the field.

- **Above** - This option places the label above the field.



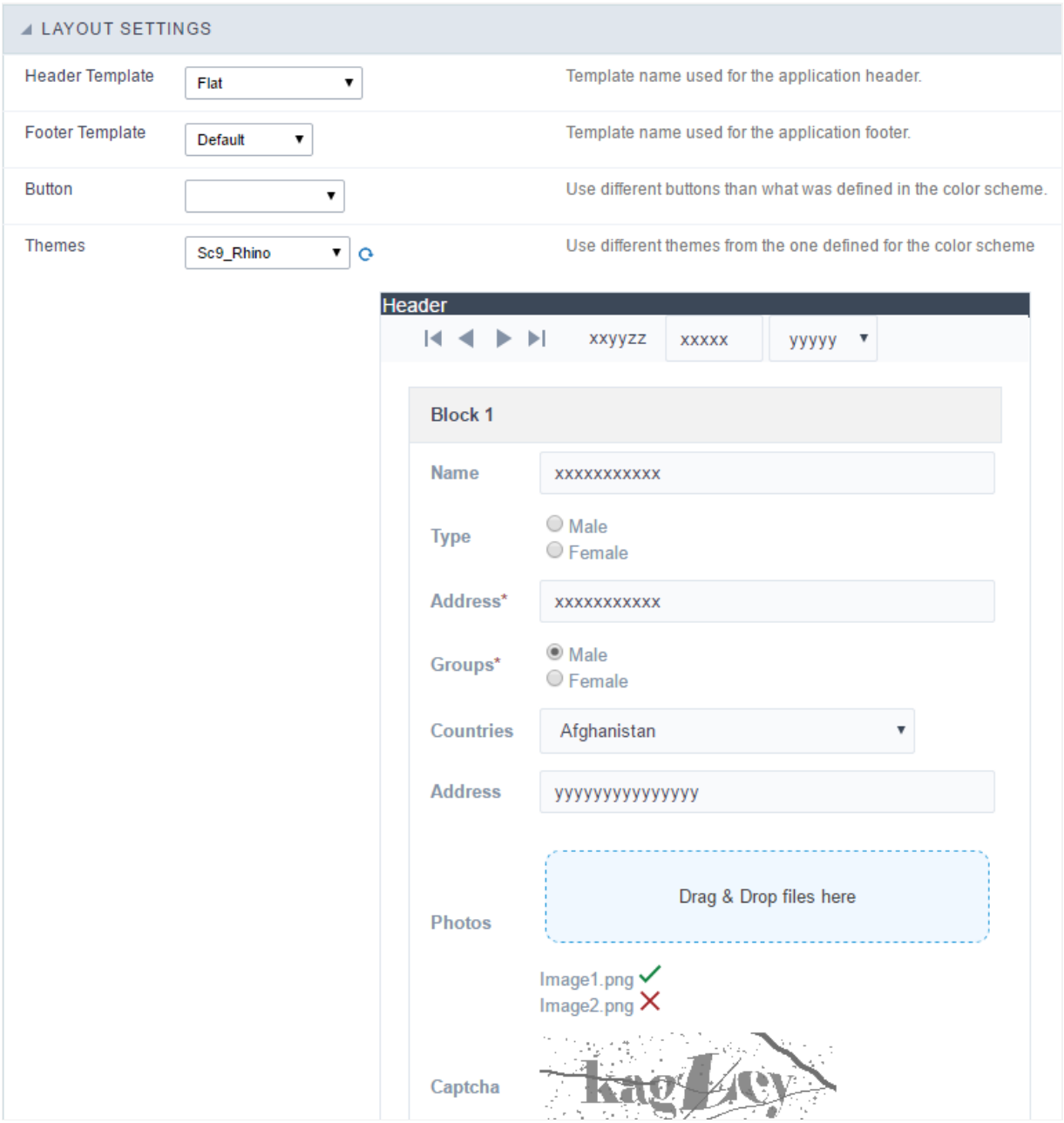
- **Below** - This option places the label below the field.

Next Block

Set the position of the following block relating to the current one. #### Border Color
The border Color for the block. #### Border Width
The border Width for the block. #### Block Width
The width for the block. #### Block Height
The Height for the block. #### Cell Spacing
The Cell Spacing in the block. #### Collapse
It enables the option to close the block.

Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

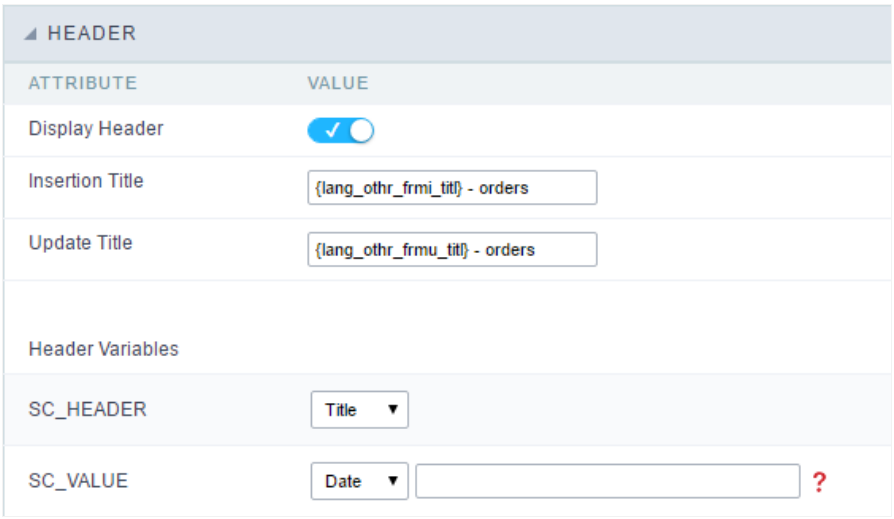
Allows choosing the button theme for the application.

Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

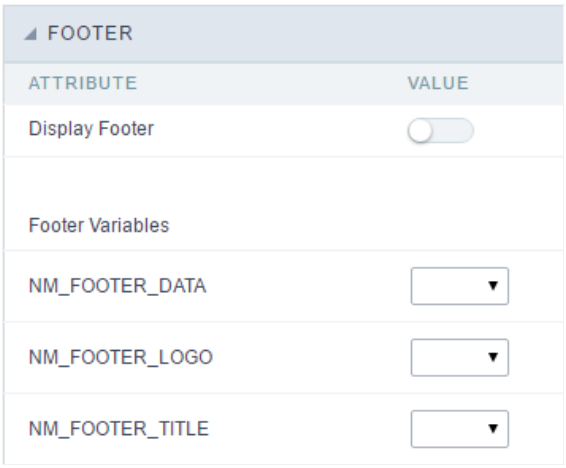
- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the

icon “**Choose Image**”, and you still can upload new images by using the button “**Upload**”. .

- **Value**: It displays the content of the text input. You can inform static texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option “**Field**”, it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system’s date in “yyyy/mm/dd” format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon “**Choose Image**”, and you still can upload new images by using the button “**Upload**”. .
- **Value**: It displays the content of the text input. You can inform static texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

Related Links 🔗

Related Video ▷

Form Evetns Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application’s CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event’s PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Form

First non-formal event executed, being triggered before the assembly of SQL and HTML.

It works as a preparation for an application, since it is possible among other things, various manipulatives, performs validations and changes a connection with a macro **sc_change_connection**, for example.

See below an example of how you use it.

No example below, or **form_orders** will be altered according to the user accessing the system.

If variable [usr_login] is equal to **admin**, or user will have access to or form with the possibility of inserting and excluding a registration.

If a variable has some other value, or user still has access to the form, therefore, only with a possibility of altering a registry.

Code example

```
if ([usr_login] == 'admin') {

   sc_apl_conf ("form_orders", "insert", "in");

   sc_apl_conf ("form_orders", "remove", "activated");

} else {

   sc_apl_conf ("form_orders", "insert", "off");

   sc_apl_conf ("form_orders", "remove", "off");

}
```

onNavigate - Form

In the form, this event is always executed when performing a navigation, through the **navigation buttons**, **navigation by page** or the option **go to**.

onScriptInit - Form

These formulas are executed before the application starts. At this moment, the application locals variables are not available.

onLoadRecord - Form

This event is executed before the form is loaded. In this moment all the applications variables are available.

onLoad - Form

This event it executed before the form is loaded. In this moment all the applications variables are available.

onRefresh - Form

This event occurs when the form application is reloaded, is possible to reload the form based on fields (select, radio, checkbox) or using the toolbar's Refresh button

onValidate - Form

This event is executed when the form is submitted to the server, using the buttons: "add", "change" or "delete"
In this event, it is allowed to perform validations before the form data is sent to the database.

In the example below, using the form_orders application, we will validate the freight field, where if the value assigned to the freight field is greater than 30, an error message will be triggered to the user and the form submission will be cancelled.

Code example:

```
if({freight} > 30){
  sc_error_message("Shipping value is too high");
}
```

onValidateFailure - FailureForm

This event occurs when submitting the application in OnValidate, has errors generated by macros sc_error_exit, sc_error_message, or error detected by scriptcase (errors related to Database).

onValidateSuccess - Form

This event occurs after submitting the form and executing the onValidate event, so if the validation is successful and contains no errors, generated by the sc_error_exit or sc_error_message macros, the onValidateSuccess event is executed.

onBeforeInsert - Form

This event occurs after form validation (onValidate and onValidateSuccess), when clicking the Insert button, and before executing the SQL command to add the record to the database.

Example: We check the user privileges table before inserting the record, if he does not have privilege, we should send an error message.

```
sc_lookup(priv_ins," select priv_ins from tb_privileges where login = [var_login] ");

if({priv_ins [0][0]} != 'YES'){

  sc_error_message(" you do not have privileges to perform this operation ");

}
```

onAfterInsert - Form

This event occurs after clicking the Insert button on the form. The form will execute the onValidate, onValidateSuccess and onBeforeInsert events, and right after executing the SQL command to insert the database record, it will execute the onAfterInsert event.

Example: After the inclusion of a record in the database, we want to store the inserting operation in a log table.

sc_exec_sql(" insert into log values ([glo_user],'target table','insert',{key_field}) "); //insert the user id that was stored in a session variable [glo_user] and the newly added registry key {key_field}.

onBeforeInsertAll - Form

This event occurs after clicking the Insert button of the **multiple record form**. The form will execute the onValidate and onValidateSuccess events, then it will execute the **onBeforeInsertAll** event before starting the execution of the SQL commands to insert the records into the database.

Note: This event is triggered only once, regardless of the number of lines inserted in one click on the Insert button.

Example: We check the user privileges table before inserting the record, if he does not have privilege, we should send an error message.

```
sc_lookup(priv_ins," select priv_ins from tb_privileges where login = [var_login] ");

if({priv_ins [0][0]} != 'YES'){

  sc_error_message(" you do not have privileges to perform this operation ");

}
```

onAfterInsertAll - Form

This event occurs after clicking on the Insert button of the **multiple record form**. The Form will execute the onValidate, onValidateSuccess and onBeforeInsertAll events, and then it will execute the **onAfterInsertAll** event after execute all the SQL commands to insert the records into the database.

For example, after entering the items of an invoice, (tb_itens_nf), in a multiple-record form, include in the "parent" table (invoice_file) the total value of the products.

```
sc_lookup (tot, "select sum (item_value) from tb_itens_nf where num_nf = {num_nf}");

{tmp_tot} = {tot [0] [0]};

sc_exec_sql ("update tb_itens_nf set total_itens = {tmp_tot} where num_nf = {num_nf}");
```

onDeforeUpdate

This event occurs after form validation (onValidate and onValidateSuccess), when clicking the Save button, and before executing the SQL command to Update the record in the database.

Example: We are checking the user privileges table before updating the record, if it does not have such privilege, we send an error message.

```
sc_lookup(priv_upd ," select priv_upd from tb_privileges where login = [var_login] ");

if( {priv_upd [0][0]} != 'YES'){

  sc_error_message("You do not have privileges to perform this operation");

}
```

onAfterUpdate - Form

This event occurs after clicking the form's Save button. The form will then execute the onValidate, onValidateSuccess and onBeforeUpdate events, and right after executing the SQL command to insert the database record, it will execute the onAfterUpdate event.

Example: After updating a record in the database, we want to store the update operation in a log table.

sc_exec_sql(" insert into log values ([glo_user],'target table','update',{key_field}) "); //insert the user id that was stored in a session variable [glo_user] and the newly added registry key {key_field}.

onBeforeUpdateAll - Form

This event occurs after clicking the Save button on the **multi-record form**. The form will then execute the events onValidate and onValidateSuccess, and then execute the event **onBeforeUpdateAll** before starting the execution of SQL commands to update records in the database.

Note: This event is triggered only once, regardless of the amount of rows updated in one click on the Save button.

As examples we can see in the example below, we consult a user privileges table before updating the record, if it does not have such privilege, we send an error message.

```
sc_lookup(priv_upd ," select priv_upd from tb_privileges where login = [var_login] ");

if({priv_upd [0][0]} != 'YES'){

sc_error_message(" you do not have privileges to perform this operation ");

}
```

onAfterUpdateAll - Form

This event occurs after clicking on the Save button of **the multiple record form**. The Form will execute the onValidate, onValidateSuccess and onBeforeUpdateAll events, and then it will execute the **onAfterUpdateAll** event after execute all the SQL commands to update the records into the database.

Example: We can update the total value of an invoice (tb_invoice) after update its items (tb_invoice_items) in a multiple records form.

```
sc_lookup (tot, "select sum(value_item) from tb_invoice_items where num_nf = {num_nf}");

{tmp_tot} = {tot [0] [0]};

sc_exec_sql ("UPDATE tb_invoice SET total_items = {tmp_tot} WHERE num_nf = {num_nf}");
```

onBeforeDelete - Form

This event occurs after form validation (onValidate and onValidateSuccess), when clicking the Delete button, and before executing the SQL command to delete the record in the database.

As examples we can see in the example below, we consult a user privileges table before deleting the record, if it does not have such privilege, we send an error message.

```
sc_lookup(priv_del," select priv_del from tb_privileges where login = [var_login] ");

if({priv_del[0][0]} != 'YES'){

  sc_error_messagem(" you do not have privileges to perform this operation ");

}
```

onAfterDelete - Form

This event occurs after clicking the form's Delete button. The form will then execute the events onValidate, onValidateSuccess and onBeforeDelete, and then, right after executing the SQL command to delete the record in the database, it will execute the event onAfterDelete.

Example: After deleting the record we can easily create logs, according to the code below:

sc_exec_sql(" insert into tb_log values([glo_user],'exclusion','customer_table',{code}); //we are storing the literals 'exclusion' and 'customer_table', the session variable [glo_user] and the field variable {code} containing the deleted content of the code field.

onBeforeDeleteAll - Form

This event occurs after clicking the Delete button on the **multi-record form**. The form will execute the onValidate and onValidateSuccess events, then it will execute the **onBeforeDeleteAll** event before starting the execution of the SQL commands to delete records in the database.

Note: This event is triggered only once, regardless of the amount of lines removed in one click on the Delete button.

As examples we can see in the example below, we consult a user privileges table before deleting the record, if it does not have such privilege, we send an error message.

```
sc_lookup(priv_del," select priv_del from tb_privileges where login = [var_login] ");

if({priv_del[0][0]} != 'YES'){

sc_error_message(" you do not have privileges to perform this operation ");

}
```

onAfterDeleteAll - Form

This event occurs after clicking no button Exclude form multiple records. Or the form will execute the onValidate, onValidateSuccess and onBeforeDelete events, and the logo will execute all the SQL commands to exclude the records not the data bank, it will execute or the **onAfterDeleteAll** event.

Note: This event occurs only once regardless of the quantity of records excluded at a time.

Ex. In an order item form (multiple records), we want to delete the order if all screen items are deleted. Just in the event OnAfterDeleteAll use a block similar to the one shown below:

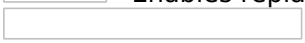
```
sc_exec_sql ("DELETE FROM orders WHERE order_id = {order}");
```

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

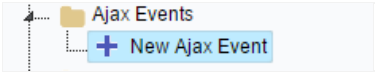
> Click Here to view the Scriptcase hotkeys documentation.

Form Ajax Events

OnClick

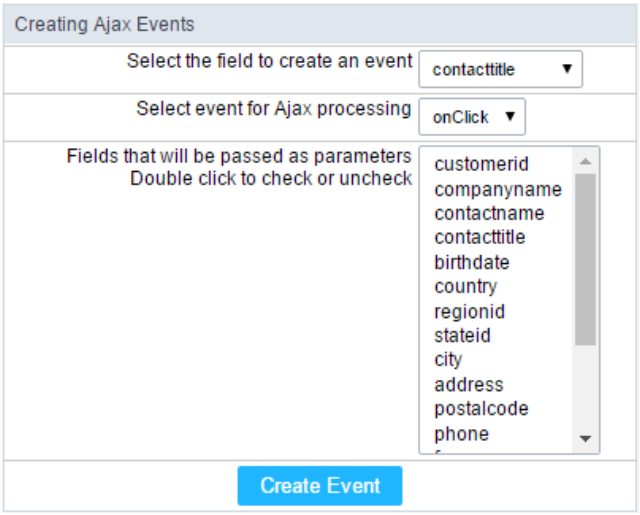
The ajax event OnClick is executed when the field that it's based on is clicked.

- Creating a new ajax event



Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

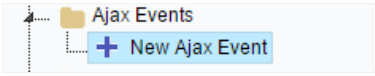
Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

OnChange

The ajax event OnChange is executed when the value of the field that it's based on is modified.

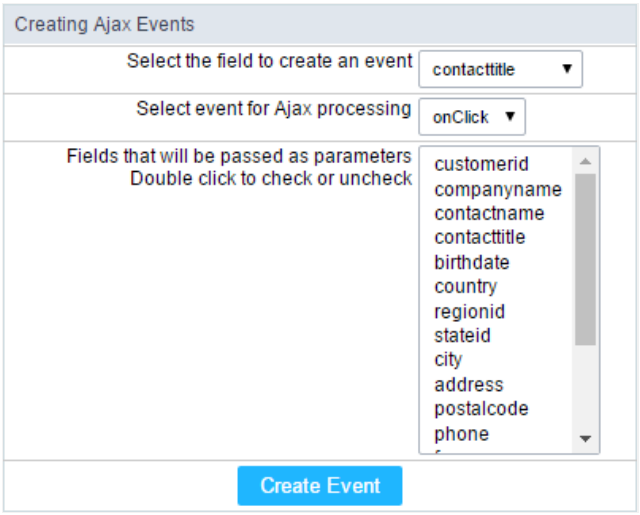
NOTE: The **radio field** is not compatible with this type of event.

- Creating a new ajax event



Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

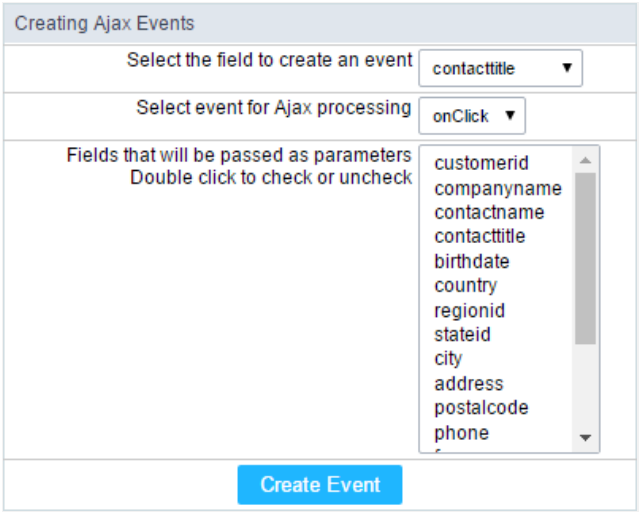
OnBlur

The ajax event OnBlur is executed when the focus is removed from the field that event is based on.

- Creating a new ajax event

Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

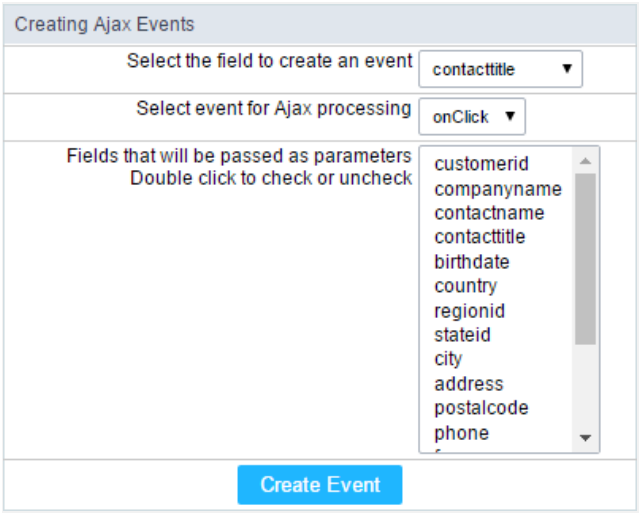
OnFocus

The ajax event OnFocus is executed when the field that it's based on is applied with a focus.

- Creating a new ajax event

Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.

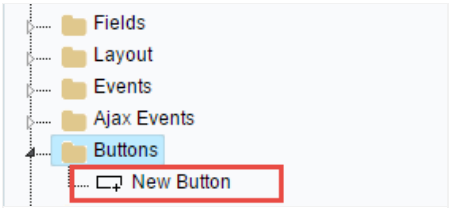
Choose an event that run the ajax Defines which event will be added to the field.

Related Links

Related Videos

Form Buttons Settings

In addition to the applications buttons, you can also create new manual buttons. These buttons will be placed on the application toolbar.

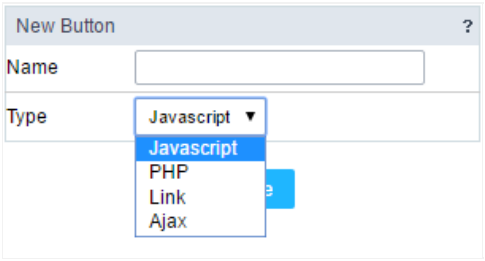


Creating new buttons

Creating a button

To create a new button, click the “new button” and type a name and the button type.

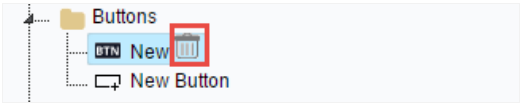
Form button types: (JavaScript, PHP, and Ajax).



Form button types

Deleting a button

To delete a button click on the icon next to the name of the button in the application menu (recycle bin).



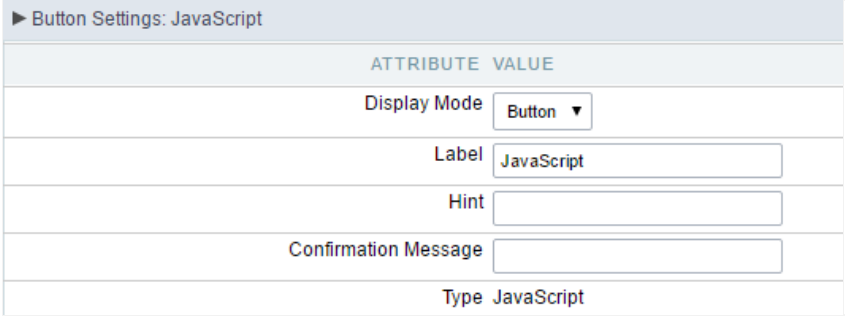
Deleting a button

JavaScript

Display Mode

You can configure the display mode of the javascript button in Image, Button or Link.

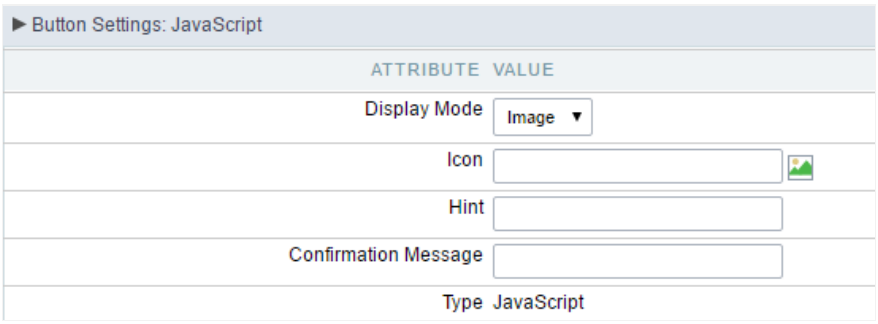
Button



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

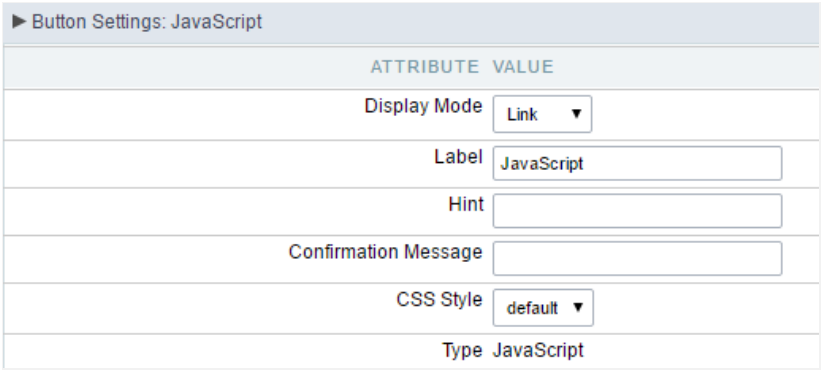
Image



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

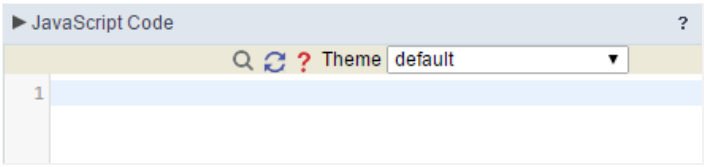
Link



Setting up Javascript Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Code Block



Javascript button coding block.

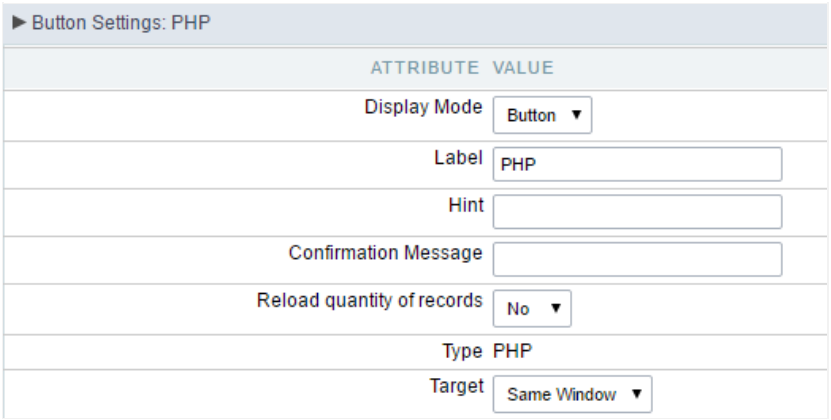
In this block, only JavaScript is accepted.

PHP

Display Mode

You can configure the display mode of the PHP button in Image, Button or Link.

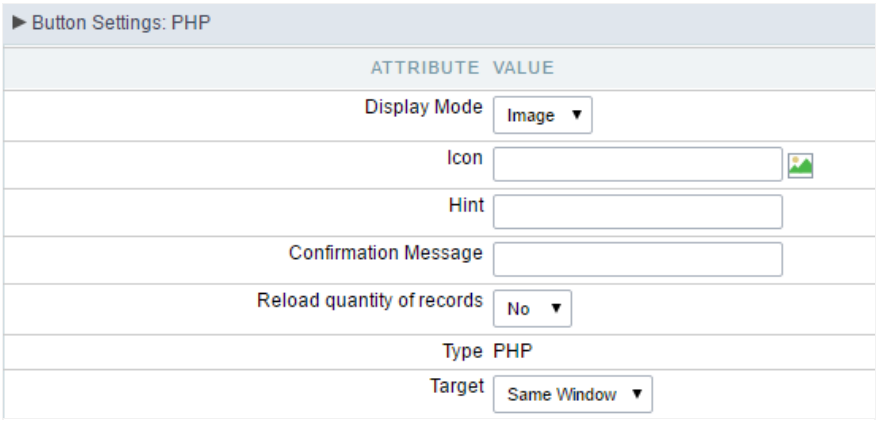
Button



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

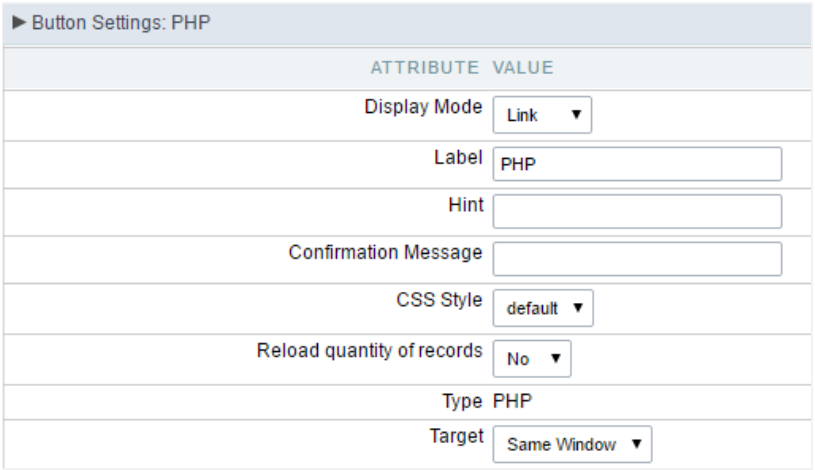
Image



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

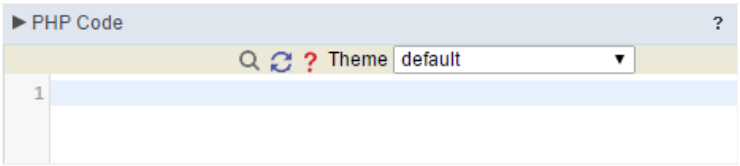
Link



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

Code Block



*Ajax button coding block. *

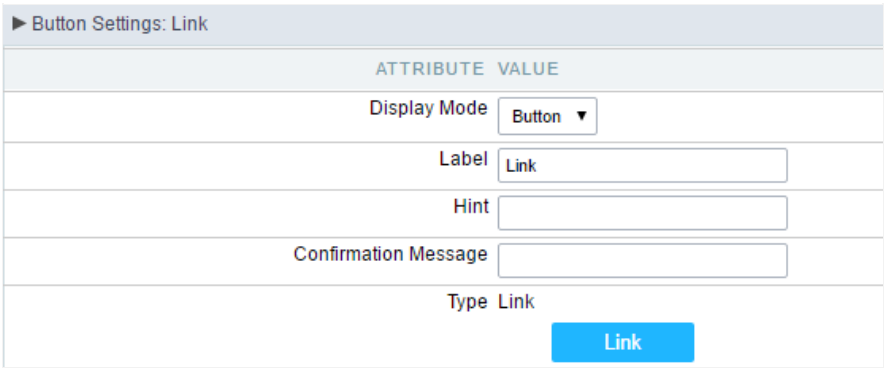
In this block, you can use macros, PHP code and JavaScript.

Link

Display Mode

You can configure the display mode of the link button in Image, Button or Link.

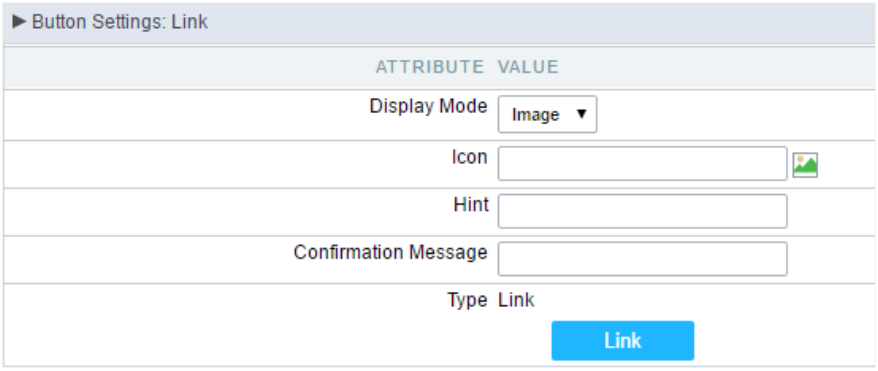
Button



Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Image

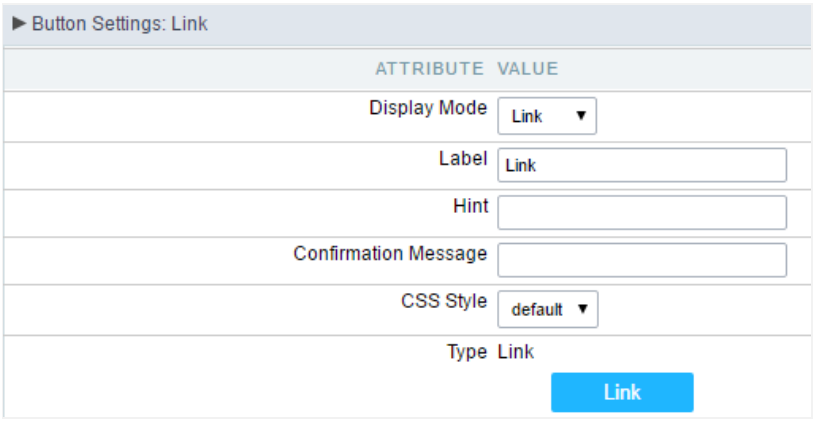


Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.

Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Link

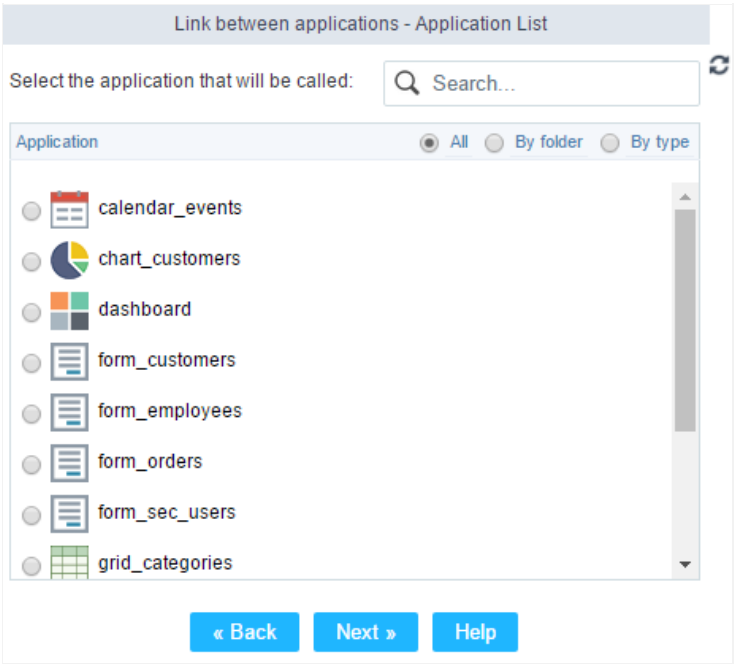


Setting up link Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Setting up the Link

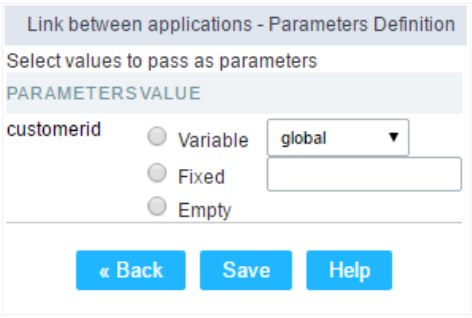
- Selecting the applications



Choosing the application for the button link.

You should select an application to be called from the button link.

- Link Parameters



Choosing the parameters for the button link.

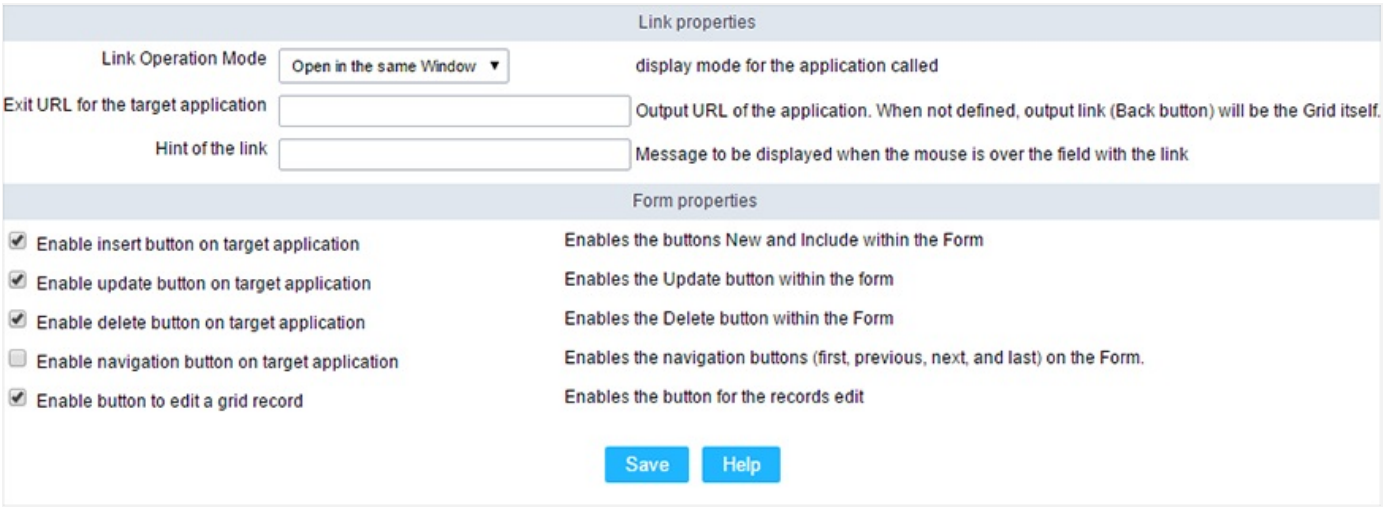
Field Allows you to use an existing field from the current application as a parameter for the link.

Variable Allows you to use a global variable from the current application as a parameter for the link.

Fixed Allows you to inform a fixed value as a parameter for the link.

Empty No value will be passed as a parameter for the link.

- Link Properties (Grid)



Configuring the properties for the link button when the destined application is a Grid.

Link Operation Mode How the link will open (Open in the same Window, Open in another Window, Modal).

Exit URL for the target application URL or an application that be redirect to when exiting the Grid application.

Initial Mode Allows you to define the initial mode of the grid application (Search or Grid).

Number of Lines Allows you to define the amount of lines displayed in the Grid.

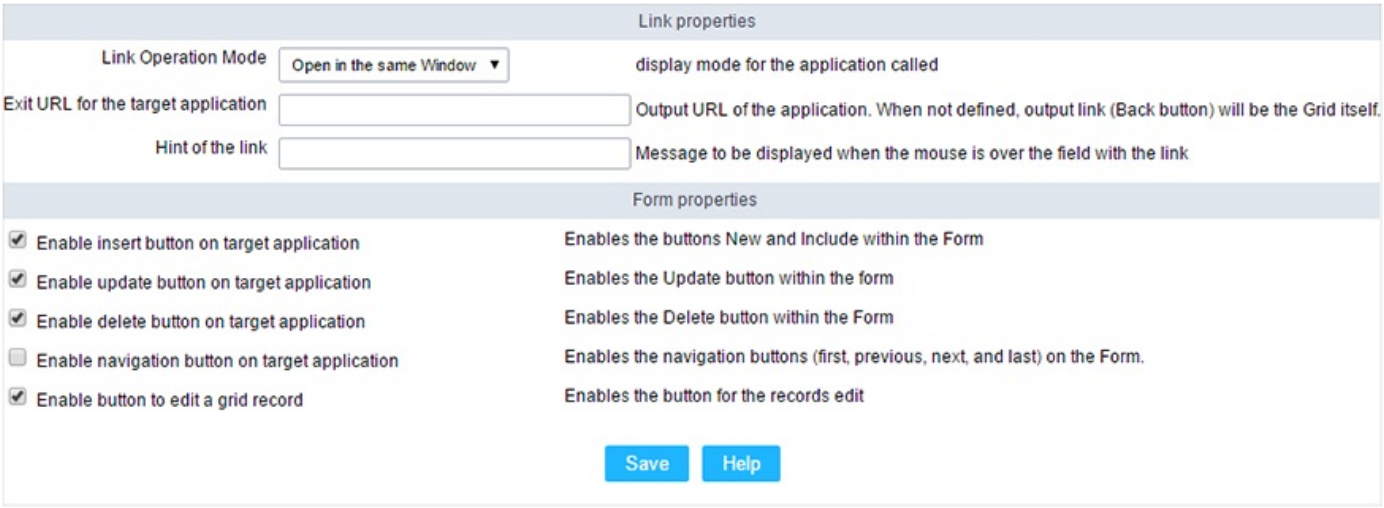
Number of Columns Allows you to define the amount of columns displayed in the Grid.

Paging Enable the paging in the Grid.

Display Header Enable the Grid Header.

Active Navigation Buttons Enable the navigation button (First, Back, Next and Last) in the Grid.

- Link Properties (Form)



Configuring the properties for the link button when the destined application is a Form.

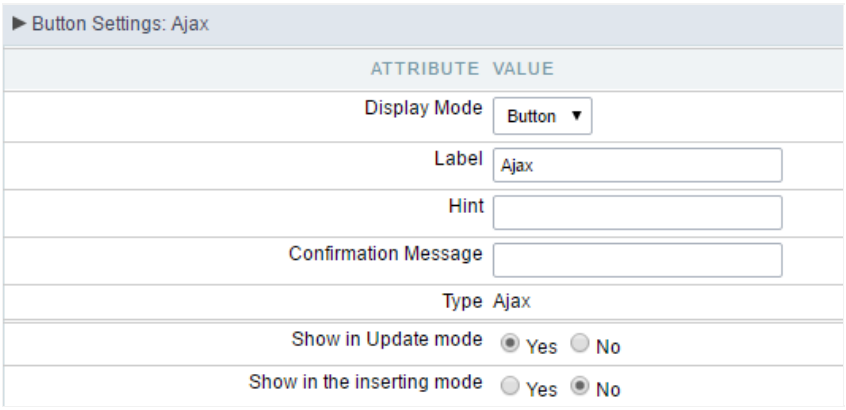
Link Operation Mode	How the link will open (Open in the same Window, Open in another Window, Modal).
Exit URL for the target application	URL or an application that be redirect to when exiting the Form application.
Enable insert button on target application	Enable the “New” button in the Form Application.
Enable update button on target application	Enable the “Update” button in the Form Application.
Enable delete button on target application	Enable the “Delete” button in the Form Application.
Enable navigation button on target application	Enable the navigation button (First, Back, Next and Last) in the Form.
Enable button to edit a grid record	Enable the buttons that allow you to edit the records of a Grid

Ajax

Display Mode

You can configure the display mode of the Ajax button in Image, Button or Link.

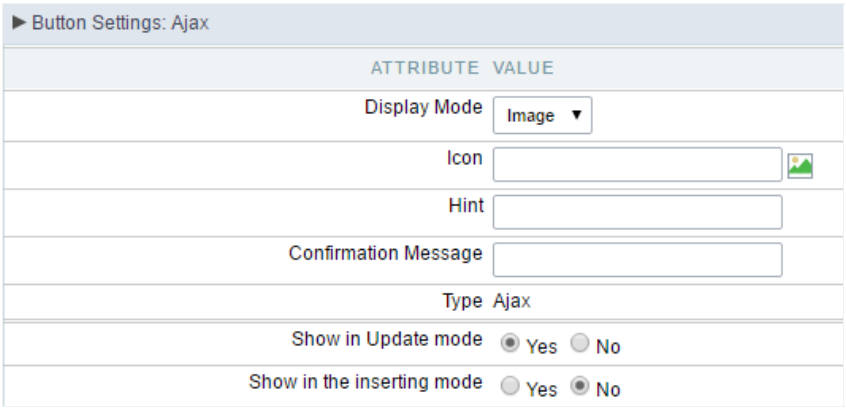
Button



Setting up Ajax Button.

Display Mode	You can select the display mode for the Ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

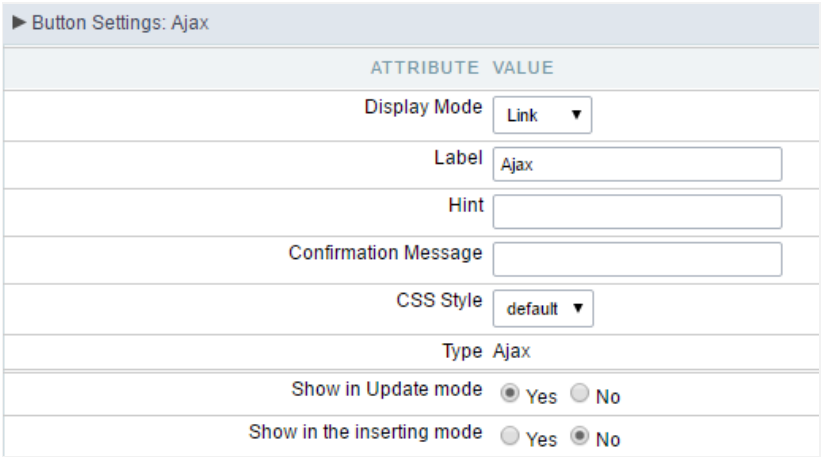
Image



Setting up Ajax Button.

Display Mode	You can select the display mode for the Ajax button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

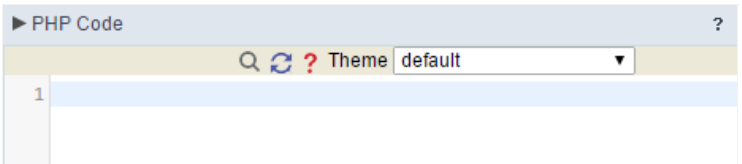
Link



Setting up Ajax Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Code Block



Ajax button coding block.

In this block, you can use macros, Ajax code and JavaScript.

Related Links 🔗

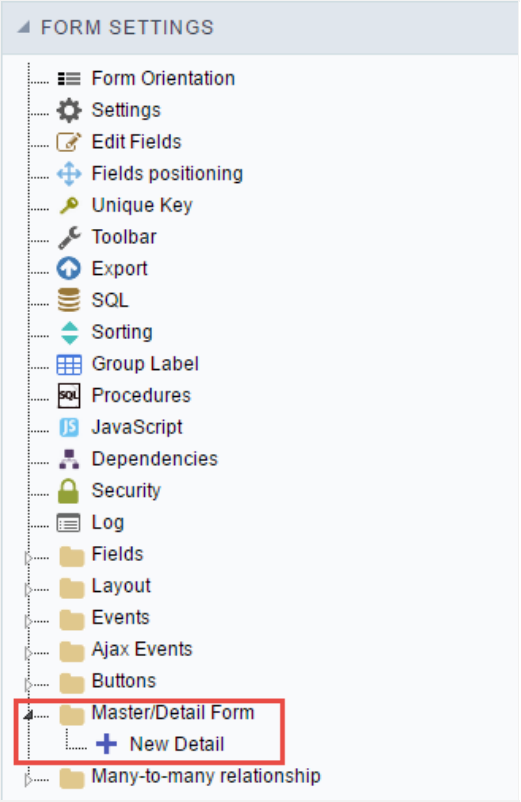
Related Videos ▷

Master/detail form

This option allows you to create forms to update, on a single screen, tables with 1:N relationship.

New detail

In the application menu, you can create a new connection by clicking on the “New detail” option from Master/detail folder.

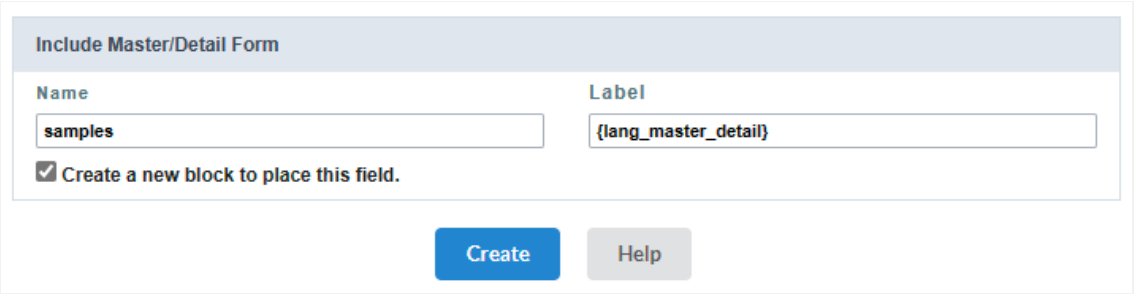


Creating a new detail.

Include name

Interface to set the **Name** and **Label** of the link. It will be displayed in the master application as a field. The name cannot contain spaces in lower special characters.

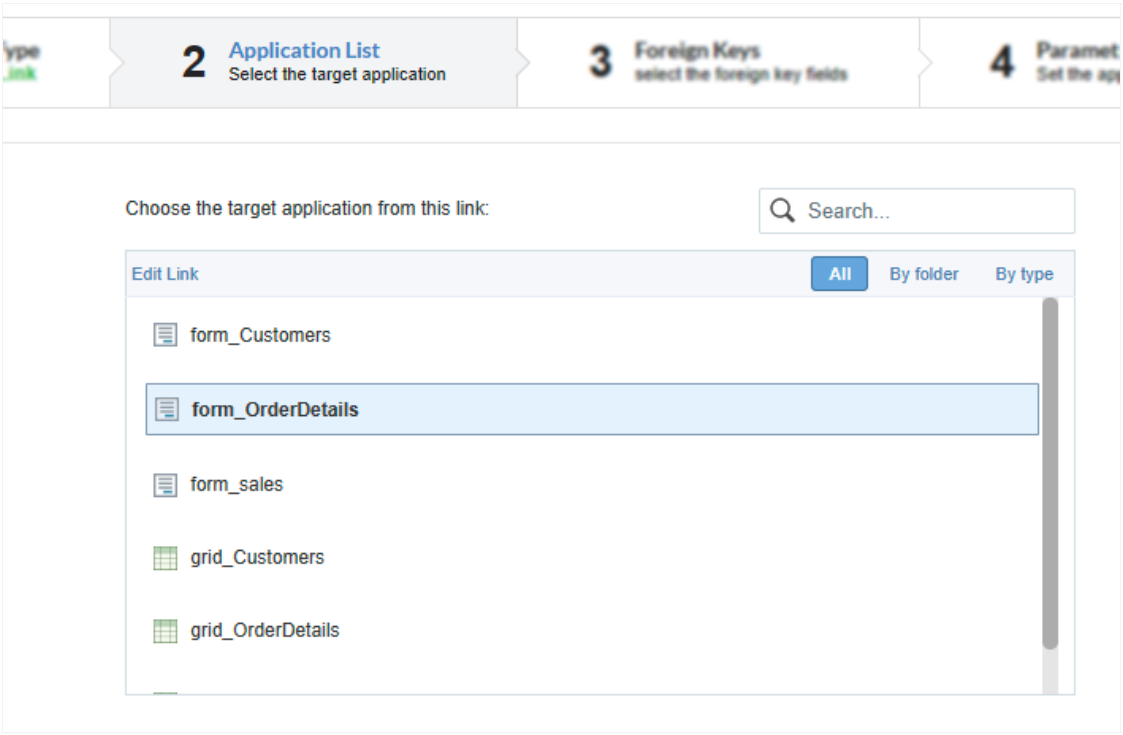
By checking the option Create a new block to place this field, the application will position itself in the new block automatically.



Interface to set link name and label.

Application list

Only the form types: Editable Grid or Editable Grid View can be used as an application detail. You must create the form detail application before you create the Master/Detail link, because you will need to select the application during the link creation.



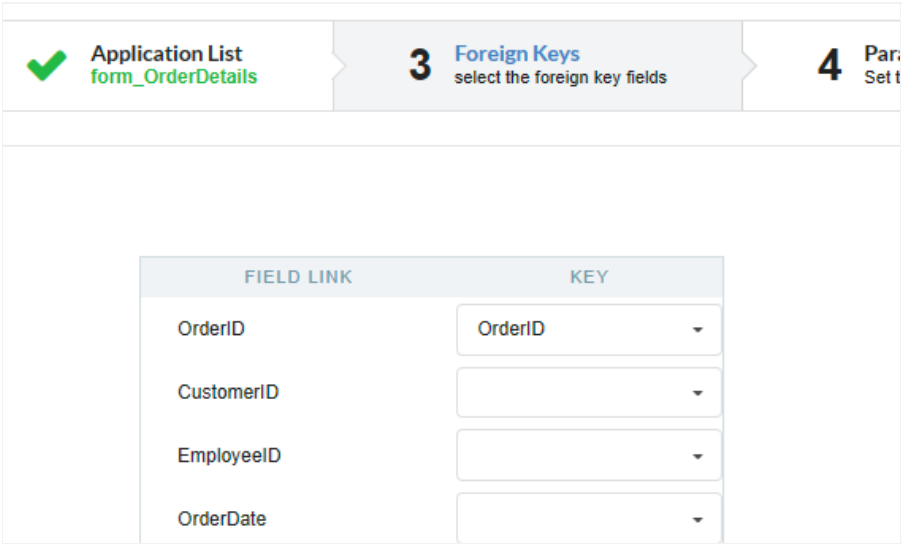
Interface to configure the detail application.

Foreign Keys

This step is enabled when you create the master/detail between form type applications. Here will be set the foreign key that links the two tables in the database.

In the left column the all fields of the master application are listed and in the select field, on the right, the list of all fields of the selected application to the Detail.

The developer must bind the fields that are foreign key in the master table. The other fields must remain blank.



Interface to configure the detail application.

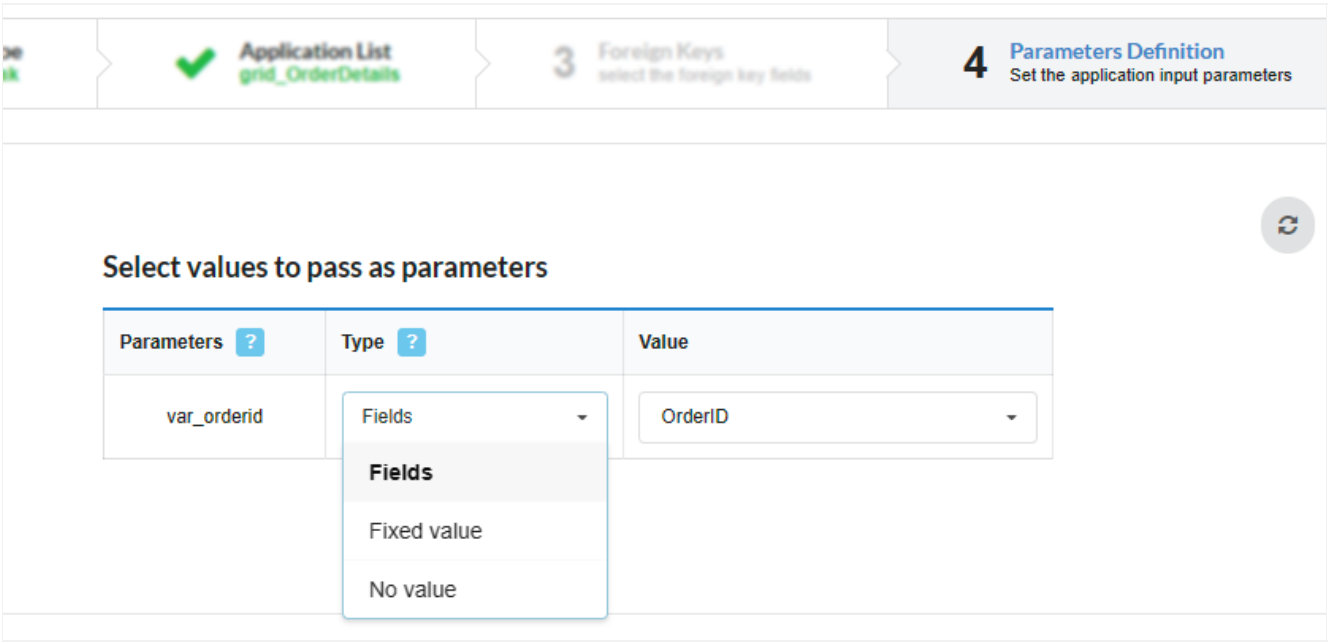
Parameters definition

In this step, the global variables defined in the detailed application for parameter passing will be listed.

For masters/details created with grid applications, defining a parameter is essential to display data in the detail related to the master being displayed.

Without this parameter, all the data from the grid used in the detail will be displayed.

Click here to access the documentation related to creating parameters in the grid.



Interface to configure the link parameters.

In the **Types of Parameters** we have three options, they are:

Fields

This option the connection will use the destination application field to make the record call according to the query application field.

Fixed value

This option allows the user to define a fixed value that will be used for the next application call.

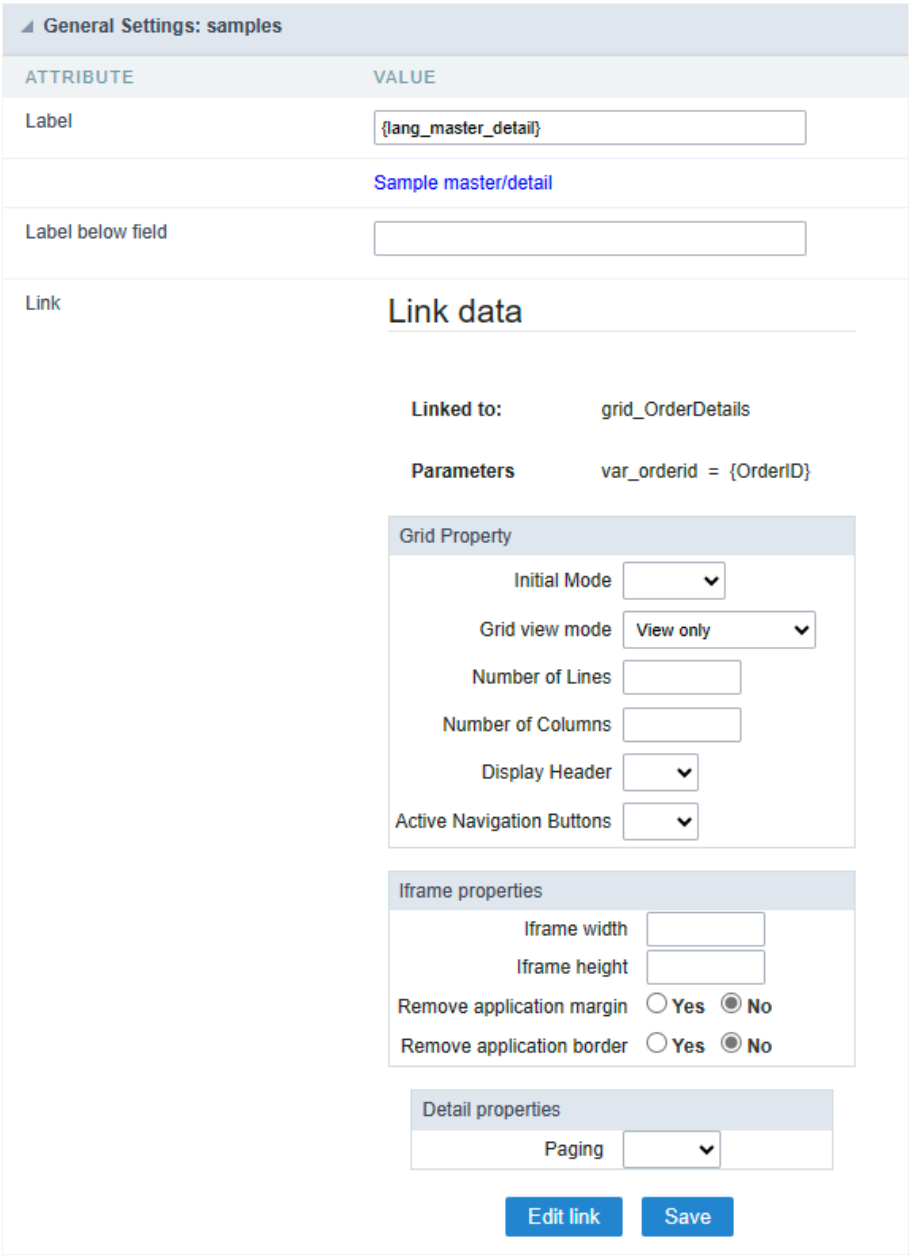
The use of variables is not allowed.

No value

This option allows you to create the connection without the need to send any parameters to the next application.

General Settings

After the link creation, you can change some settings by clicking on the link and selecting the properties option.



Label

Define the name shown in the application. This attribute accept text or language variable.

Label below field

Define the text that will be shown below the detail. This Attributo to accept text or language variable.

Grid Property

- **Initial Mode** - Defines whether the detail application will be initiated by the **search** module or the **grid** module.
- View only -
- **Number of Lines** - Defines the number of lines that the application will show at the beginning.
- **Number of Columns** - Defines the number of columns that the application will show at the beginning.
- **Display Header** - Defines whether the application header will be displayed.
- **Active Navigation Buttons** - Defines whether the application Navigation Buttons will be displayed.

This configuration block is only displayed in master detail between a grid and a form.

Form properties

Within this option you can select the buttons that are going to be displayed in the detail form.

- Enable insert button on target application
- Enable update button on target application

- Enable delete button on target application
- Enable navigation button on target application
- Maintain the WHERE clause in the target application

This configuration block is only displayed in master detail between two forms

Iframe properties

This option allows you to configure the iframe height and width (in pixels) where the detail form is going to be displayed.

- **Iframe width** - Sets the width of the iframe
- **Iframe height** - Sets the height of the iframe
- **Remove application margin** - Remove the application margin inside the iframe
- **Remove application border** - Remove application borders inside the iframe

Detail properties

- **Form Type** - The option “Form Type” only works to change applications already configured with horizontal orientation, used within the detail. E.g. Multiple Rows or Editable Grid. **This attribute is only displayed in links between two forms.**
- **Paging** : Allows you to select your form detail paging. The options are:
  - **Partial** : This option will set page layout as partial, with a limited records per page, according to the quantity informed within the option below “Lines per page”
  - **Total** : This option will set the page to display all records in one page.
- **Lines per page** : Here you can set the amount of records that are displayed in the detail form, you will need to set that just if paging is partial.**This attribute is only displayed in links between two forms.**
- **When deleting** : This option allows you to choose how the form detail dependencies will be treated in a deletion. **This attribute is only displayed in links between two forms.** The options are:
  - **Warn when there are dependencies** : Displays an error message if there is dependency on the detail form record that is being deleted from the master form.
- **Delete dependencies without warning** : Deletes all existing dependencies in the detail form without warning the user.

Related Links 🔗

Related Videos ▷

Many-to-many Relationship

Allows you to configure automatic update link tables in N:N relationships

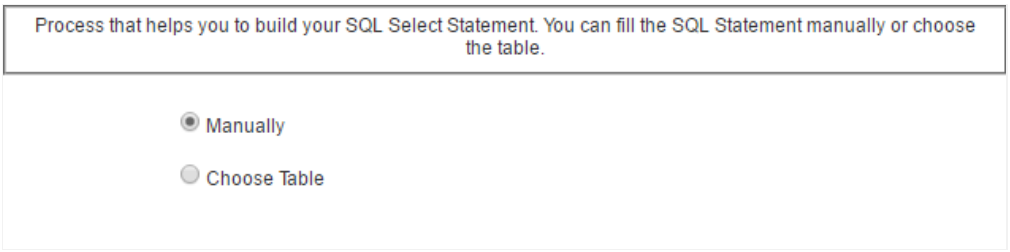
Identification

This link will be handled as a field within the form application. We have to set the field name, data type, label and define if we will use a specific connection for this link. If decide to you use a specific connection there will be a combo box with for you to select according to the connections available in the project.

Creating field to update link table.

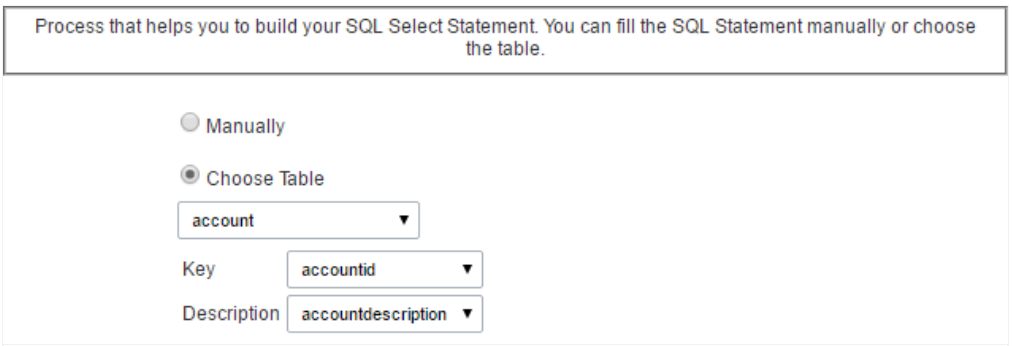
Grid Information

In this step you need to inform the data source, this connection will be available for the user to select (lookup). It allows you to choose if the select command is manually informed or if it will be based on a table.



Informing the select command.

If you check the option “Choose table” there will be 3 combo boxes for you to select according to the connections tables available in the project: table, key field and description.

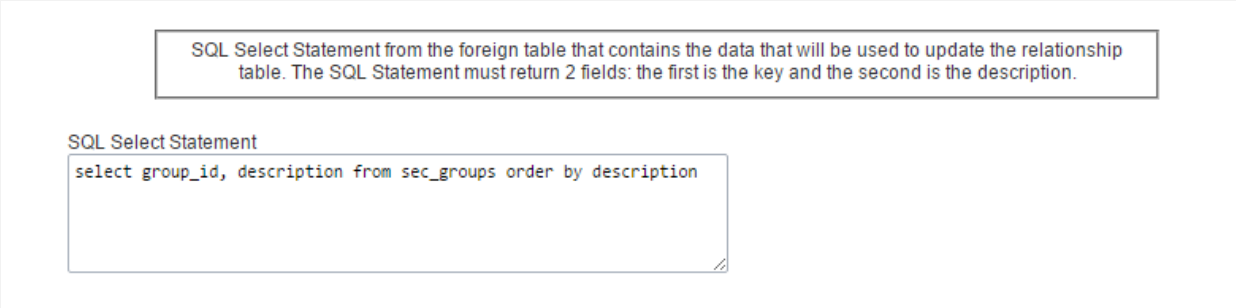


Informing the select command based in a table..

- **Key** : Value that will be stored in the table.
- **Description** : Value that will be displayed in the update field.

SQL Select Statement

Select command informed by user or created based on a table, it is responsible for displaying the selection field contents.



Select command.

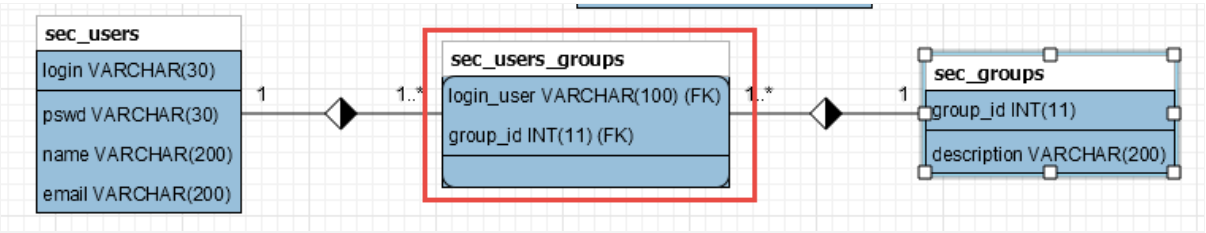
Lookup display

This option sets the display for the update field

Field object type.

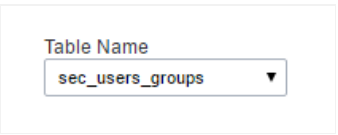
- **Display key and description** : If you want to display the code and the description informed in the select command.
- **Display Title** : Displays attribute title (see attribute).
- **Separator** : When the option Display key and description is checked as yes you must define a separator between the code and the description.
- **Object** : Object field type that will be displayed in the update form. The types are: Select, Radio, Check box and Double Select.
- **Check and Uncheck All** : Option to Check and Uncheck all records (available only for when Check box object is selected).
- **Columns** : This option sets the number of columns (available only when the objects Check box or Radio is selected)
- **Height** : This option sets the object height (available only when the objects Select or Double Select is selected).

Link Table



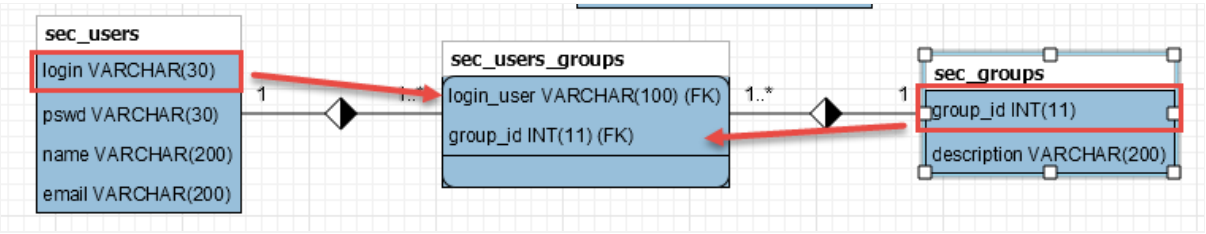
Link table.

This option sets the relationship table that will be updated.



Update table selection.

Relationship Keys



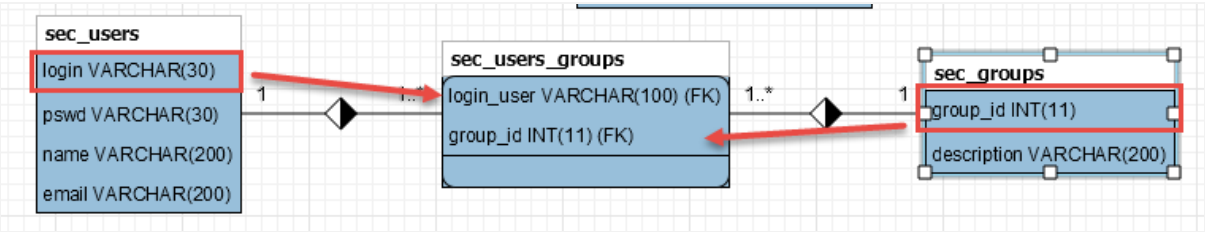
link key

This option sets the application fields that are related with the link table fields.

Field	Key
login_user	customerid
group_id	regionid

Setting values for the fields in the update table.

Lookup Field



Lookup field.

This option sets the field from the lookup table that is related to the link table.

Lookup field
group_id

Defining the foreign key relationship table..

Link Attributes

This option sets the values that are recorded in the fields of the link table that are not foreign key.

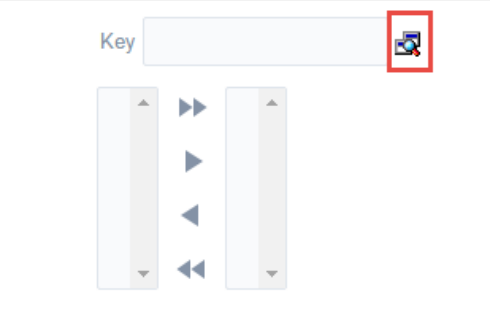
Field	Attributes
login_user	customerid

Setting the assignment field.

- **Attributes** : It is written a value without the need for user data entry, it is an application field itself.
- **Data entry** : It is written a value reported by the user, it is necessary to create a field and relate it by the select.

Search

Available only for Double Select object



Search in the double select field.

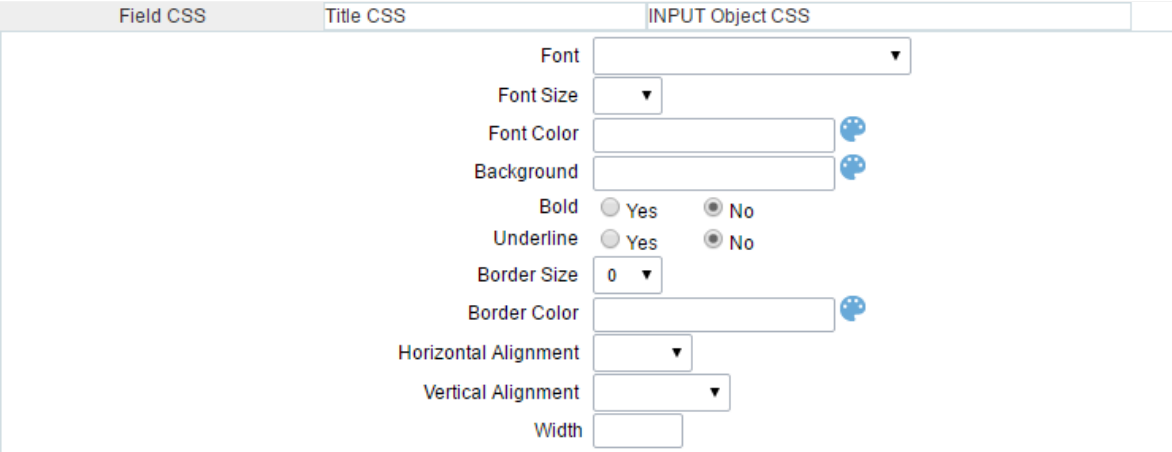
This option sets a filter for the link field.

![Search settings for the link field.][relacao_n_n_info_filtro_config] *Search settings for the link field.*

- **Limit of Records** : Maximum number of records returned by the search.
- **Initial State** : this option sets whether the form field load come Filled or empty.
- **Search Fields** : This option sets whether the search will be displayed the code and/or Description.

General Display Settings

This option allows you to configure all general display settings from from field title and object.



Display settings.

Help Configuration

This feature allows it to be documented instructions for use of the application generated, that is, help end users to better understand the business rules of the system and the best way to operate it.

Help Configuration

- **Attributes**
- **Help Description** : This option allows the inclusion of a text that will be displayed when the user to position the mouse over the field.
    - **Help Type**:
        - **Pop-up** : When choosing the type pop-up, help icon is displayed next to the field; that when clicked, will display what was reported in the attribute Describes Help.
        - **Hint** : When you pass the mouse cursor over the field, will help described in the field Describe Help.

- **Text** : Next to the field will appear a text containing the help that was described in the field Describe Help.

Related Links 🔗

Related Videos ▷

Form Search Overview

The chart application offers the following types of filters for the user:

Advanced Filter

The Advanced Filter is another powerful filtering tool in Scriptcase. It offers more detailed options to refine query results. With the Advanced Filter, you can select columns, comparison operators, and values to create custom filter conditions. Additionally, you can combine multiple conditions using logical operators like “AND” and “OR.” The Advanced Filter is ideal when you need more precision in your queries, allowing you to define complex criteria to find the desired data.

Filter Builder

Allows the end user to create filters during the use of the application, combining different conditions and logical operators (AND/OR) to create more robust filters and generate more detailed reports.

Below is an example of how the filter builder works.

The user can add several fields or nested conditions to combine the use of logical operators.

Dynamic Filter

Allows the end user to build filters during the use of the application by adding the fields that should compose the filter according to their needs, within the same logical operator.

Below is an example of how the dynamic filter works.

The user must select the logical operator that will be used for all filter fields: All conditions (AND) or any condition (OR).

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos

Form Search Settings

With this interface, you can define general options of the Search Form.

Search configuration Interface.

Search Criteria

Allows to select the logical operator **AND** or **OR** to define the criteria of the search;

Display Condition

Gets the condition of the search available for the user to choose one. He can select “AND” or “OR” in a Combobox.

Use auto-complete in the fields

Automatically turns the field into an autocomplete according to the existing values in the database. If the user chooses **Yes**, the autocomplete will enable automatically in all inputs that contain a relationship. If the user decides **No**, so no autocompletes will be displayed. Otherwise, the option selected is **Defined in the field** it’ll keep the settings for each field individually.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Ads
- Search with Ads Advanced and Dynamic search
- How to do a Pivot Table with Filter
- Pivot Table with Filter
- Highlight option

Related Videos

Form Search Criteria Settings

With this interface, you can configure the conditions available for each field of the Search form.

Search configuration Interface.

We can see the fields list on the left combo. On the right, the list of options for filtering the selected field. To select an option, click on one of them (Equal to, Beginning with, Contains, etc.) and then the button On/Off. The arrows, on the right, allows altering the order of the fields.

For the Date type fields, you can define special conditions for the search, accessing the field configurations, and editing the Special Conditions Settings.

Below the list are the buttons to enable the selected options:

- **On/Off**: Enables or disables the field or the option chosen.
- **All**: Marks all fields or options.
- **None**: Unmarks all the fields or options.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refine search and Highlight option
- Search with blocks
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos

Form Dynamic Search Settings

On this screen, the general behavior of the dynamic filter in the application will be defined.

Filter type

Defines the type of filter that will be used in the application.

Filter Builder

Allows the end user to create filters during the use of the application, combining different conditions and logical operators (AND/OR) to create more robust filters and generate more qualified reports.

Below is an example of how the filter builder works.

The user can add multiple fields or nested conditions to combine the use of logical operators.

Dynamic filter

Allows the end user to build filters during the use of the application by adding the fields that should compose the filter according to their needs, within the same logical operator.

Below is an example of how the dynamic filter works.

The user must select the logical operator that will be used for all filter fields: All conditions (AND) or any condition (OR).

Use the ENTER key for

This attribute defines the behavior of the ENTER key when the filter screen is being displayed.

Form Filter Build Settings

Enable/Disable

This attribute defines whether the user can disable fields or nestings in the filter builder, keeping them disabled until the application is reloaded.

Example of the disable button

Display filter condition

Defines whether the filter condition will be displayed while the user builds their filter.

When enabled, the condition is assembled as the user adds the filter’s conditions and logical operators. When disabled, this condition will not be displayed.

Modal Filter

Defines the location where the filter builder will open in the application.

When enabled, the filter will open in a modal. If disabled, the filter will be displayed in a div below the application’s toolbar. See below for examples of both behaviors.

Opening in a modal

Div below the toolbar

Select Fields for Filter Build in Form

On this screen, the fields that will be available for use in the dynamic filter must be defined.

In the **left column**, all application fields and the **Search** field are displayed, which helps in locating the fields. In the **right column** are the fields selected to be displayed in the filter.

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

- - Move all fields to the right.
- - Moves only selected fields to the right
- - Moves only the selected fields to the left.
- - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the button.

> To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning

Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

> Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

Select Fields

On this screen, the fields that will be available for use in the dynamic filter must be defined.

In the **left column**, all application fields and the **Search** field are displayed, which helps in locating the fields. In the **right column** are the fields selected to be displayed in the filter.

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

- - Move all fields to the right.
- - Moves only selected fields to the right
- - Moves only the selected fields to the left.
- - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the button.

> To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning

Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

> Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

Related Link

- Dynamic Search

Related Video

- Macro sc_where_filter
- Highlight option

Quicksearch

Quicksearch is an option that allows to search data in various fields of the application by using the text box in the toolbar.

Quick Search Settings

In the quick search settings you can find the following options:

Quicksearch layout

Simplified: Standard quicksearch view. This option displays the input where we must inform the terms to be searched for.

Extended: In this layout, in addition to the simplified layout options described above, it is possible to select the columns where a search will be carried out and the criteria to be selected.

The criteria must be selected in the option **Research Criteria**, they are:

Contains, Equal beginning, Exactly equal, Does not contain, Different, Greater than, Greater than equal, Less than and Less equal

Highlight results

Highlights the result of the research performed.

Button within the search area

Defines the placement of the search button, whether it will be in the text area or outside it.

Option Enable -

Option Disabled -

Display combobox in the simplified layout

When active, display combobox with the fields selected in the **Quicksearch extended fields** option

Watermark

Watermark that will be displayed in quicksearch. A lang or a fixed text can be used.

By default, we use the lang: {lang_othr_qk_watermark}

Quicksearch width

Width in pixels of the Quicksearch input.

Settings for Individual Search by field

This configuration defines the fields that are available in the quicksearch combobox, when activating the Extended Layout option, in the general settings

Fields for individual search

Use fields displayed in the application: When selecting this option, only the fields displayed in the application will be available in the combobox.

Define fields manually: Allows the developer to select the fields that will be available in the quicksearch combobox.

Search settings in the ‘All Fields’ option

This option defines the fields that will be used in quicksearch, regardless of their configuration. That is, all fields selected in this option will always be available in the search.

Search in All Fields

All Fields added to the fields of the Individual Search : In addition to the fields selected in this option, the fields selected in the **Quicksearch extended fields** option will be available in the quicksearch search.

Only those selected for All Fields: Only the fields selected in this option will be available in the search through quicksearch

Search Criteria

Defines quicksearch search criteria.

When checking the options, the combobox with the criteria will be displayed

These are the options available as a quicksearch search criterion.

- **Contains**
- **Equal beginning**
- **Exactly equal**
- **Does not contain**
- **Different**
- **Greater than**
- **Greater than equal**
- **Less than**
- **Less equal**

Related Link

- Quicksearch

Related Video

- Quicksearch and Highlight option
- Macro sc_where_filter

Form Search Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

How to synchronize table fields with a form

Anytime you add/delete a field or change its data type, you have to synchronize your form application with the connected table to apply that changes.

For more information, click here.

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid		
Procedure		
Chart		
Form		
Control		
Calendar		

Application	Table Fields	Virtual Fields
Search		
PDF Report		

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven’t manage fields.

Text

General Settings

This type of field allows the developer to create quickly inputs to insert and update data, where the final user can inform its data to be allocated in its database.

Type of Data

Defines the type of the field in the application. In this case we should select **Text**.

Label

Defines the title that will be displayed in the field when the application is executed. The terminology used in the interface is fundamental to the system have a good usability, whe should use names and familiar terms to the final user of the application, instead of using terms from the system.

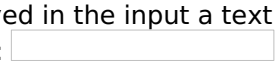
For example, this text field that has the name **customerid**, the client would have a much better understanding of the functionality of the field when we define the label as **Customer Name**.

Not only a fixed text, the **Label** attribute allows the use of langs to define the field's title, making it possible to use your application in a multi language project.

Label below field

Defines the message to be displayed below the field.

Watermark

Informing a text to the **Watermark** it will be displayed in the input a text as an example that can be informed in the field. The result after set will be this: 

Initial Value(type)

Allows the initial definition to the field when the form is in insert mode. It is possible to choose between two options:

Defined Value: When this option is selected, the Initial Value attribute will be available, where we should inform the field's initial value. For example, my initial value is **Arlindo**, when a new register is inserted, the field **Seller Name** will be initialized as Arlindo.

System Data: When this option is selected, the initial value will be the actual date of your computer's system.

Amount of Characters

Allows to set the width of the text field's input that varies with the amount of characters informed. Although, if the amount of characters typed are greater than the setting, the text will be pushed to the left, to keep the maximum amount of characters as defined.

Show HTML Content

When this option is active every HTML, CSS and JavaScript content that are in the database will be displayed with the main value.

Validation Image

When this option is active, a image will be displayed next to the informed field if the field is according to the settings defined by the developer.

In the example below, the field was set to receive at least 5 characters, see what happens when informed

only 4 characters:

However if informed 5 or more characters the field will be displayed as:

Password Field

When this option is active, the text field will be converted to the format used in password fields. For example:

View password characters

By enabling this option, a button will be displayed in the password field so that the password is displayed when clicked.

Save Variable

Allows to save a session variable(global variable) with the field's value, to be used in others applications.

For example, in the login form the username can be saved in session and displayed on the header of others applications.

Variable Name

In this attribute we should define the name of the session variable, active in the previous item, that will receive the field's value.

We should inform only the variable's name, - **var_rating**.

The method to use its value is global variable.

Field Mask

Defines the field mask. There are two types of mask described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

For example, it is possible to set the mast to display a telephone number:

It will be show with this format on runtime:

It is also possible to set the field mask like those examples:

Field mask examples:

Telephone number

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678

Field	Mask Input	Typed Value	Formatted Value
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Validate with mask

When this option is active it is possible to insert the data with the mask set in the **Field Mask** option.

Complete to the left

When this option is used, the value defined will be added on the left of the information inserted in the field if the value is lower than the maximum set in the **Maximum Size** option.

Field size in database

Defines the field's size related to the size set in the database. This value is already set automatically by default when the application is generated.

Hidden Field

When this option is active, the field will be hidden in the application on runtime.

Label Field

When this option is active, the field will be altered to only a label where the info will be displayed, where updates or inserts will not be possible.

Save HTML tags

When this option is active, it allows to HTML tags in the field to be inserted with the data, instead of being interpreted.

Text input in JavaScript

When this option is active, it will be show every JavaScript content inserted in the database with its main value.

> This option can only be used using the **Editable grid view**.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disable Field

Allows the developer to disable the field, so the user can not type a value according with the option defined by the developer.

The available options are: **No** - Initial value of the attribute, this option does not disable the field.
Update Mode - This option only disables the field when editing existing registers.
Insert Mode - This option only disables the field when inserting new registers.
Insert / Update Mode - This option disables the field when editing exciting registers or when inserting new registers.

HTML Type

HTML object used to display the field in the form.

SQL Type

Informs the type of the field in the database.

Field Behavior

Use autocomplete:

Field automatically turns into autocomplete according to existing values in the database.

Use Select2

Uses the new component for data selection, allowing searches within the select comboBox.

Width for the Select2

Sets a width for the area in the Select2.

Amount of characters

Sets the amount of characters to start the search.

Amount of rows

Sets the maximum number of rows to list the search result.

Width:

Defines a width in pixels for a result box.

Search options:

Defines the validation that will be made to fetch the search result.

Start equals to: Will return the records with the same start value as in the database.
Any part: Will return the records when exist the character in any part of the record.
End equals to: Will return the records with the same final value as in the database.

Position between values:

Defines the position that objects will be displayed.

Text Between Values

Text that will appear when using a filter condition between two values.

OnChange Submit:

Submit search on this field changing.

Show Condition:

To show or not the search condition. It only works if the search has at least one condition.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Macro sc_where_filter
- Highlight option

Related Videos

Integer

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Label:

It sets the title that will be displayed in the field when you run the application. The terminology used in the interface is fundamental for your system has a good usability. In this case, we recommend you to use names and terms familiar to the end user of the application, instead of using terms originated in the system.

Label below field

Defines the message to be displayed below the field.

Watermark

Informing a text for the **watermark** it will be displayed on the input text with an example of what can be reported in the field. The result after that will be configured:

Initial value (type)

Allows setting an initial value for the field when the application is in inclusion. You can choose between two options:

Pre-set value: When you select this option, the Initial Value attribute will be made available in the place where we inform the initial value of the field.
For example, my initial value is **Arlindo** when inserting a new record the **seller_name** field will start with Arlindo.

System date: When you select this option, the initial value will be the current system date of your computer.

> If you select the type system date, it will be not necessary to fill in the initial value attribute.

Amount of characters

It allows you to set the width of the input text field according to the amount of characters. However, if the quantity entered is greater than the set for the characters, the text will be pushed to the left, in order to ensure the maximum amount of characters set in the option of **Values formatting**.

Validation Image

When you enable this option, an image will be shown next to the field informing whether the field is in accordance with the settings of *minimum size* and *Maximum* size (as you can see in the images below) defined in the Formatting value option.

In the example below, the field was set to receive at least 5 characters, see what happens with the image when it receives only 4 characters:

However, if the value inserted has 5 or more characters the the image will change according to example bellow:

Use slider:

It displays a slider component in the field. So you can increase or decrease the value sliding the cursor. You can also customize the increment value, if it increments the value 1 by 1, 2, 5, 10... N.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Validate with Mask

By enabling this option you will be able to enter data according to the mask that was configured in the option **Field Mask** and Scriptcase will validate it.

Record Variable

Allows you to record a session variable with the value of the field ([global variable] [var_glob]), to be used in other applications.

Example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable Name

In this option you must set the session variable name, enabled in the previous item, which will receive the value of the field.

You need to inform only the variable name, for example: **var_seller**.

The recovery of the value is made as [Global Variable][var_glob]{:target='blank'}.

Field size in Database

This option sets the size of the field relative to the size that is configured in the database. By default this value is already configured automatically when the application is generated.

Hidden Field

This option when enabled will hide the field inside the application at the time of execution.

Label Field

By enabling this option, the field will be changed to only one label where the information will be displayed, so it is not possible to make changes or inserts in the field configured as label.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disable field

Allows you to disable the field, making it impossible for the user to enter a value according to the developer-defined option.

The available options are:

No - initial value of the attribute, this option does not disable the field. **Update mode** - This option disables the field only when editing the records. **Insert mode** - This option disables the field only inserting new records. **Update/Insert mode** - This option disables the field in both editing and inserting new records.

HTML Type

HTML object used to display the field in the application.

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:

Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

```
SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}
```

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading

Assigned value	Description in Lookup
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**

- **Font** : Allows to choose the font type, that will be applied to the application field title.
- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Advanced search with a predefined initial value

Related Videos

- Dynamic Search
- Advanced search with relative periods
- Refine search and Highlight option
- Search with blocks
- Quick Refined Advanced and Dynamic search
- Pivot Table with Filter
- Macro Set_Where_Filter
- Highlight option

Decimal

General Settings

Decimal field Configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Field Behavior

Use autocomplete:

Field automatically turns into autocomplete according to existing values in the database.

Use Select2

Uses the new component for data selection, allowing searches within the select comboBox.

Width for the Select2

Sets a width for the area in the Select2.

Amount of characters

Sets the amount of characters to start the search.

Amount of rows

Sets the maximum number of rows to list the search result.

Width:

Defines a width in pixels for a result box.

Search options:

Defines the validation that will be made to fetch the search result.

Start equals to: Will return the records with the same start value as in the database.
Any part: Will return the records when exist the character in any part of the record.
End equals to: Will return the records with the same final value as in the database.

Position between values:

Defines the position that objects will be displayed.

Text Between Values

Text that will appear when using a filter condition between two values.

OnChange Submit:

Submit search on this field changing.

Show Condition:

To show or not the search condition. It only works if the search has at least one condition.

Values format

Decimal Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator, Negative sign and Negative number format.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Advanced search with relative periods
- Refined Search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Currency

General Settings

Currency field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Currency, you can currency values to the field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Currency Field Behavior Interface of the Search Configuration.

- **Use auto-complete** : The field behaves as an auto-complete according to the values existing in the database.
- **Amount of characters** : Sets the amount of characters to start the search.
- **Amount of rows** : Sets the maximum number of rows to list the search result.
- **Width** : Sets the width in pixels for the result box.
- **Search options** : Defines the validation that will be made to fetch the search result.
- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Values format

Currency Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).

    - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the title of the field.
    - **Height** : To define a height for the title of the field.
- **CSS of the Field**
    - **Font** : Allows to choose the font type, that will be applied to the application field.
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the field.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
    - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the field.
    - **Height** : To define a height for the field.
- **CSS of the Input Object**
    - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the Input Object.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
    - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the Input Object.
    - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
    - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

    - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

    - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search

Related Videos

- Advanced search with highlighting
- Define Refined Search
- Search with Filter
- Pivot Table with Filter
- Advanced search with date periods
- Refined Advanced and Dynamic search
- Search with Slicers
- Pivot Table with Filter

Date

General Settings

- **Data Type** : Define the type of field for the application. When set to Date, you can inform a date.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Watermark**: Displays a watermark in the field input.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the datatype of field in the database.

Values format

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator attribute.
- **Date separator** : Allows you to inform the separator symbol for the date.
- **Display** : Select the format of the day for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **Date**. You need to use the characters **A**, **M** and **D** that correspond to **Year**, **Month** and **Day**.
- **Use Combo-box** : Allows you to select the date using a combo-box.
  - **Year as Combo** : Allows to use the year combo to select the date.
    - **Initial Year** : First year displayed in the combo.
    - **Actual Year +** : Display the current plus the amount of years informed.
- **Month in full textual** : Displays the Month format in Full.
- **Display Calendar** : Enables the a calendar icon beside the field, this allows to select the date from a calendar with the format already setup.
  - **New Calendar** : Defines if the JQuery calendar (New Calendar) is going to be displayed or the old format.
  - **Years Limit** : Amount of years displayed in the calendar.
  - **View week number** : Displays the number of the week in the application.
  - **Additional months** : Displays the additional months of the calendar.
  - **Show Combo year and month** : Displays the year and month of the calendar in the combo box.

Field Behavior

- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Special Conditions

- **All Period** : Searches for all periods of dates.
- **Today** : Searches in today’s date.
- **Yesterday** : Searches in yesterday’s date.
- **Last 7 days** : Searches the last 7 days. Ex: ((01/01/2017 01/07/2017).
- **This month** : Searches the dates from the first day of the current month.
- **Last month** : Searches the dates from the first day of lasts month.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of

Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refine Search
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Time

General Settings

Time field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Time, you can inform a time to this field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Values format

Time Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator and time separator attributes.
- **Time separator** : Allows you to inform the separator symbol for the time.
- **Display** : Select the format of the time for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **TIME**. You need to use the characters **HH**, **II**, and **SS** that correspond to **Day**, **Hour**, **Minutes** and **Seconds**.

Field Behavior

Time Field Behavior Interface of the Search Configuration.

- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.

- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Search with Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos

Datetime

General Settings

Datetime field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Datetime, you can inform a date and time to this field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Values format

Datetime Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator and time separator attributes.
- **Date separator** : Allows you to inform the separator symbol for the date.
- **Time separator** : Allows you to inform the separator symbol for the time.
- **Display** : Select the format of the day/time for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **DATETIME**. You need to use the characters **A**, **M**, **D** , **HH**, **II**, and **SS** that correspond to **Year**, **Month**, **Day**, **Hour**, **Minutes** and **Seconds**.
- **Use Combo-box** : Allows you to select the date using a combo-box.
  - **Year as Combo** : Allows to use the year combo to select the date.
    - **Initial Year** : First year displayed in the combo.
    - **Actual Year +** : Display the current plus the amount of years informed.
- **Month in full textual** : Displays the Month format in Full.
- **Display Calendar** : Enables the a calendar icon beside the field, this allows to select the date from a calendar with the format already setup.
  - **New Calendar** : Defines if the JQuery calendar (New Calendar) is going to be displayed or the old format.
  - **Years Limit** : Amount of years displayed in the calendar.
  - **View week number** : Displays the number of the week in the application.
  - **Additional months** : Displays the additional months of the calendar.
  - **Show Combo year and month** : Displays the year and month of the calendar in the combo box.

Field Behavior

Date Field Behavior Interface of the Search Configuration.

- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Special Conditions

Datetime Field Special Conditions.

- **All Period** : Searches for all periods of dates.
- **Today** : Searches in todays date.
- **Yesterday** : Searches in yesterdays date.
- **Last 7 days** : Searches the last 7 days. Ex: ((01/01/2017 01/07/2017).
- **This month** : Searches the dates from the first day of the current month.
- **Last month** : Searches the dates from the first day of lasts month.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**

- **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links 🔗

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos ▷

Select

General Settings

Select field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Select, you can select multiple option from a combo box (Select Field).
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Select Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

  Selecting the lookup type.

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimitation.

      - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Height** : Defines the height for the select object.
    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to

a value informed manually (Title internal value).

- **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
- **Title** : Text displayed on the title of the Select object.

- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.
- **Link** : Allows to create a link to another form allowing to manipulate the list displayed on the select field. After the manipulation, the select object it updated automatically.

- **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Height** : Defines the height for the select object.
  - **Multiple Values (delimiter)**

    You can store various values for the select field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Height** : Defines the height for the select object.
  - **Multiple Values (position)**

    Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of

bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Height** : Defines the height for the select object.

- **Multiple Values (binary)**

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

Editing Lookup Configuration Multiple Values (Binary)

- **Label** : Text that will be displayed in the item list of the select field.
- **Height** : Defines the height for the select object.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  Displaying the original lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Lookup Method - Actual value

This lookup is used to list all the values in the selected field.

This lookup will apply a “distinct” to your SQL query.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Dynamic Search
- Advanced search with relative periods
- Refined Search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Related Videos

Double Select

General Settings

Double Select field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Double Select, your allowed to have multiple options selected.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Double Select Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, getting these values from the database.

Lookup Settings Display for the field. *Automatic Lookup Interface..*

- **SQL Select Statement** : Defines the SQL command that will get the values displayed on the form field. The command needs to have the following structure:

  ```
  SELECT key_field , display_field FROM table
  ```

  The value of the key_field will be stored in the table field.

- **Height** : Set the height(lines) of the field interface.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter than (;).
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**

- **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links 🔗

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos ▷

Check box

General Settings

Check box field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Check box, your allowed to have multiple options selected.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Check box Field Behavior Interface of the Search Configuration.

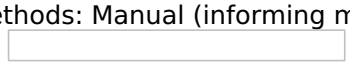
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).  *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the CheckBox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
    - **Option check all** : Displays two options on the field to check and uncheck all.

    - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Columns** : Set the amount of columns, for the list of items.
  - **Multiple Values (delimiter)**

    You can store various values for the checkBox field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Columns** : Set amount of columns, for the list of items.
  - **Multiple Values (position)**

    Stores a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

    For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

    The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

    Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the

following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Columns** : Set the amount of columns, for the list of items.

- **Multiple Values (binary)**

  Stores a diversity of information selected on the field in only one value.

  In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  To store the data, it is done an internal sum of the decimal number. See the examples below.

  Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 2 (Culture) = **3**

  Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

  Setting up Multiple Values (Binary)

  - **Label** : Text that will be displayed in the item list of the checkBox.
  - **Columns** : Allows you to inform the amount of columns, for the list of items.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Saves all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Refreshes the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Option check all** : Displays two options on the field to check and uncheck all.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio,

Select, Text, etc
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

    Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

    Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

    Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Refined Search
- Quicksearch and Highlight option
- Search Refined Advanced and Dynamic search
- Pivot Table with Filter
- Highlight option

Related Videos

Radio

General Settings

Radio field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Radio, your allowed to select one of the options listed.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Radio Field Behavior Interface of the Search Configuration.

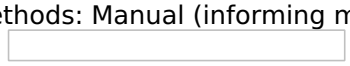
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).  *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Columns** : Allows you to inform the amount of columns, for the list of items.

    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

      - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
      - **Title** : Text displayed on the title of the Select object.

- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

  This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

  Manual Lookup Interface.

  - **Label** : Text that will be displayed in the item list of the radio.
  - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
  - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
  - **Insert Button** : Adds to the list the values informed on the fields Label and Value.
  - **Update Button** : Modifies the attributes of the selected item.
  - **Remove Button** : Remove the selected item from the list.
  - **Clear Button** : Clear the Fields.
  - **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on load lookup definition.
  - **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
  - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

    - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
    - **Title** : Text displayed on the title of the Select object.
  - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
  - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

    Displaying the original lookup.
  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Defined Search
- Quicksearch and Highlight option
- Search Refined by Advanced and Dynamic search
- Pivot Table with Filter
- Macro se_where_filter
- Highlight option

Related Videos

Text Auto-Complete

General Settings

Text Auto-Complete field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Text auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal Text for the data.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Text Auto-Complete Field Behavior Interface of the Search Configuration.

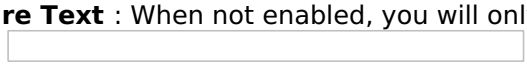
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **Case Settings** : Allows to convert the letters of the field when losing the focus. The options are:
  - **Upper Case** : All in Upper Case.
  - **Lower Case** : All in Lower Case.
  - **Capitalize first word** : Capitalizes the first letter of the word.
  - **Capitalize all words** : Capitalizes the first letter of all the words.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below.  *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value
- Quicksearch
- Dynamic Search
- Advanced search with relative periods
- Defined Search
- Quicksearch and Highlight option
- Search Refined by Advanced and Dynamic search
- Pivot Table with Filter
- Macro se_where_filter
- Highlight option

Related Videos

Number Auto-Complete

General Settings

Number Auto-Complete Field Behavior Interface of the Search Configuration.

- **Data Type** : Define the type of field for the application. When set to Number auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal number for the data.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Number Auto-Complete Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Values format

Number Auto-Complete Field Format of Values with Regional Settings.

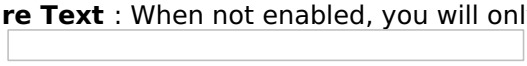
- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator, Negative sign and Negative number format.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below.  *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps

the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Related Links

- Advanced search with a predefined initial value

Related Videos

- Dynamic Search
- Advanced search with relative periods
- Quick search and Highlight option
- Search with blocks
- Pivot Table with Filter
- Highlight option

Date Comparison

Special field that allows you to change the functioning of a date type field in the summary filter, enabling the comparison of metrics over time. To use this type of field, it is necessary to have a date type field created, and later change its type to Date Comparison.

> It is only available in the advanced filter and its operation is exclusive to the query summary module.

Example of the field in the summary filter

Example of the Generated summary

When using the comparison, the summary will be assembled showing the values for the periods informed.

General Configuration

In this block are the general settings of the field, and where it will be 'transformed' into a comparative data field, simply access the settings of a data field and change the *Data* type to **Date comparison**

Data type of the field.

This attribute defines the type of field in the application. This definition implies the configuration options that will be displayed to the developer as well as the behavior of the field when running the application.

During the application creation process, Scriptcase maps the fields of the table(s) used and defines the data type of the fields according to their SQL type.

For example
Fields of type varchar in the table will be defined as text in the **Data Type** attribute.

> It is possible to change the field's data type, however it is important to take into account the **SQLType**, displayed in the General Configuration block, to define the field's data type.

Search Label

Defines the title of the column displayed in the executed application and can be defined using: Lang, Text or Global variable.

Lang

The use of lang in the attribute definition is recommended for creating multilingual systems. The langs must be previously defined using the option Traduzir aplicações or through Dicionário de Dados.

Fixed text

In this case, the column title will not be changed due to user interaction, such as changing the application language, for example.

It is a good option for systems created with just one language.

Global variable

The use of global variables allows you to change the column title according to your system business rule.

> We also provide the macro sc_label which allows you to change the column title regardless of how the label attribute was defined.

Example of a column title

Use same label used on the Grid

This option, when activated, allows you to use the title of the field defined in the grid, in this case the value defined in the **Label filter** attribute will be disregarded.

Value Formatting

Regional Settings

Allows you to apply regional date formatting settings to the field, according to the application language. When disabling the use of *regional settings*, some settings will be displayed such as: **Date separator**, **First Day** and **Display**.

Check below for more details about each configuration option.

> It is possible to change regional settings in general, allowing the use of the option and the same configuration in all your projects, according to the language used. For this configuration, access the Menu *Locales* > *Configurações Regionais*

Date separator

Defines the character that will be used to visually separate the parts of the date (day, month and year). The characters normally used are: slashes (“/”), hyphens (“-“) and periods (.).

Alguns exemplos

- **slashes (/)**: 25/10/2023
- **hyphens (-)**: 25-10-2023
- **periods (.)**: 25.10.2023

> This option will be displayed if the **Use regional settings** attribute is disabled.

First day

Defines the first day of the week to be displayed in the calendar in the date field, for selecting dates.

First day change example

In this example, the *First day* was defined as Monday, changing the order in which the days of the week are displayed in the field calendar.

> This option will be displayed if the **Use regional settings** attribute is disabled.

Display

Defines the date display format by selecting one of the listed formats.

The options are:

- dd-mm-aaaa
- mm-dd-aaaa
- aaaa-mm-dd

> This option will be displayed if the **Use regional settings** attribute is disabled.

View week number

Defines whether the number of weeks will be displayed in the date field calendar.

Example of a calendar with number of weeks

Special Conditions

Defines the special conditions that will be displayed in the date field calendar, see the example below.

Example of special conditions in the date field calendar

If no interval is enabled, the calendar in the date field will not have any special conditions.

Example of the date field without special conditions

Select Values

In this attribute, the special conditions that will be listed as an option in the application must be defined. The predefined intervals are grouped according to the period they refer to. For example, the condition *Last week* is in the tab **Week**.

In each of the tabs it is also possible to create personalized intervals using the button Add new range.

In the tab , all selected ranges will be listed, so that the display order can be changed according to the user's needs.

Sorting tab

In this tab, all selected conditions will be listed, allowing the intervals to be reordered according to the system's needs. You can also remove conditions one by one or using the button **Remove all**.

Add New Range

Clicking on *add new range* will open a screen with some options for adding new ranges. To add an interval, we must select the type (next or last) and enter the period (value).

For example, adding a range in the year tab, defined *Last* in the attribute as **Type** and reporting *2* in the attribute **Value** a range will be created: **Last two years**

> This button is available on the **Year**, **Quarter**, **Month**, **Week** and **Day** tabs.

Type

Defines whether the created interval will refer to a future period (next) or a previous period (last).

- **Next** - Defines that the created interval will refer to a future date, for example, **Next year**.
- **Last** - Defines that the range will refer to a previous date, for example, **Last Year**.

Value

Sets the amount of time for the interval.

> This field only accepts positive integer values greater than zero.

Title

Defines the text that will be displayed in the application. When creating a new range, a lang is automatically

generated, it is also possible to use a fixed text.

Include Current

This option changes how the created interval works, adding the current period to the interval.

For example, adding an interval to the year tab, defining *Next* in the **Type** attribute and entering *2* in the **Valor** attribute and checking the **Include current** option will create an interval: **Next 2 years from current**.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse

leaves the help icon.

- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

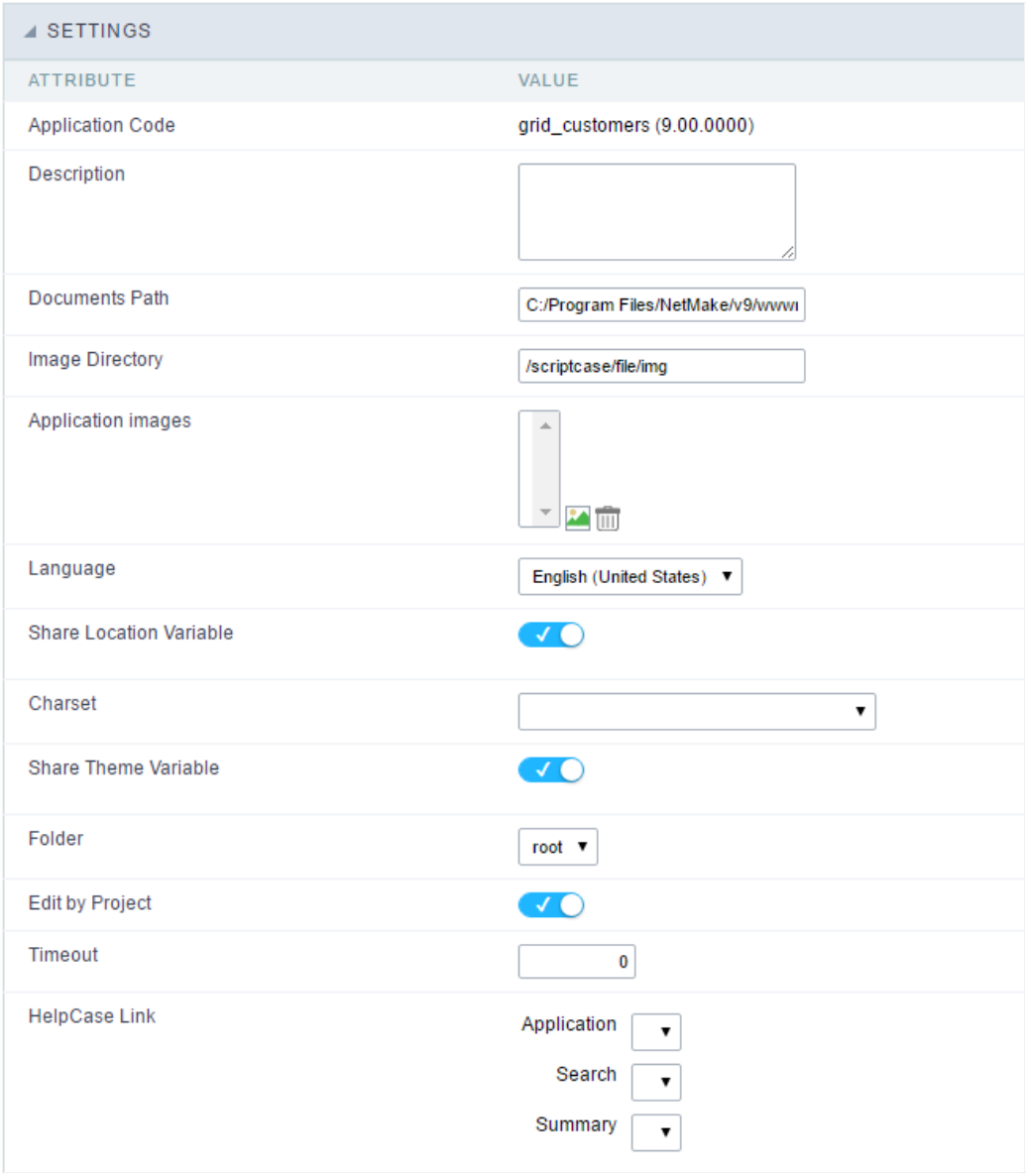
Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Application Settings (Form)

Settings

With this interface, you can set the common attributes of the app.

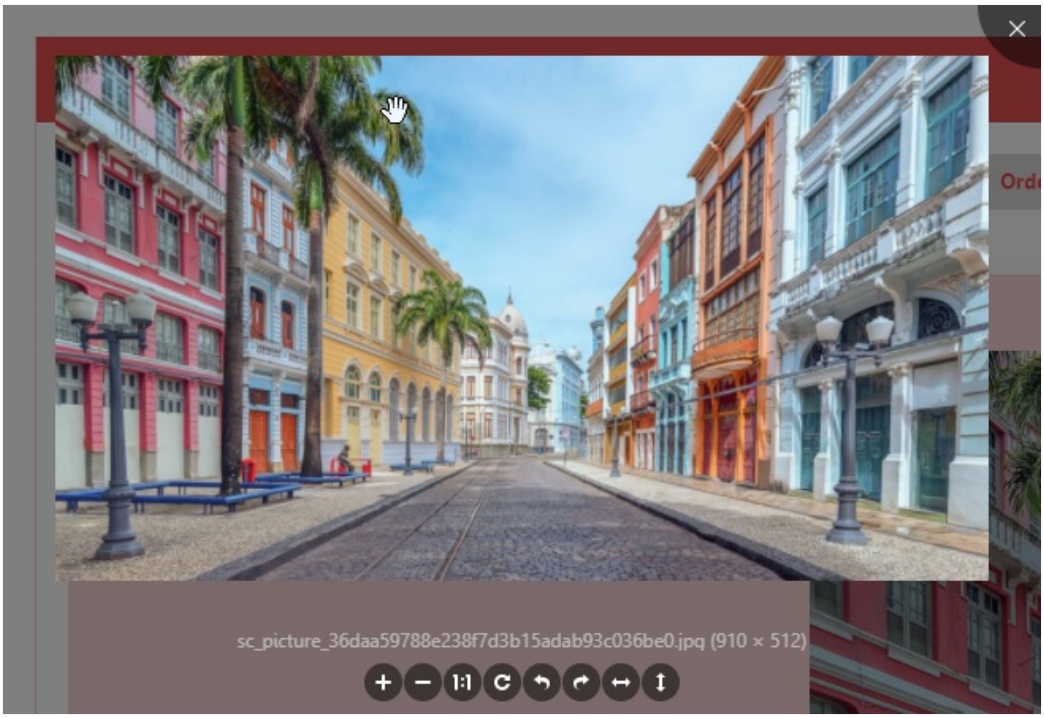


Application Settings Interface

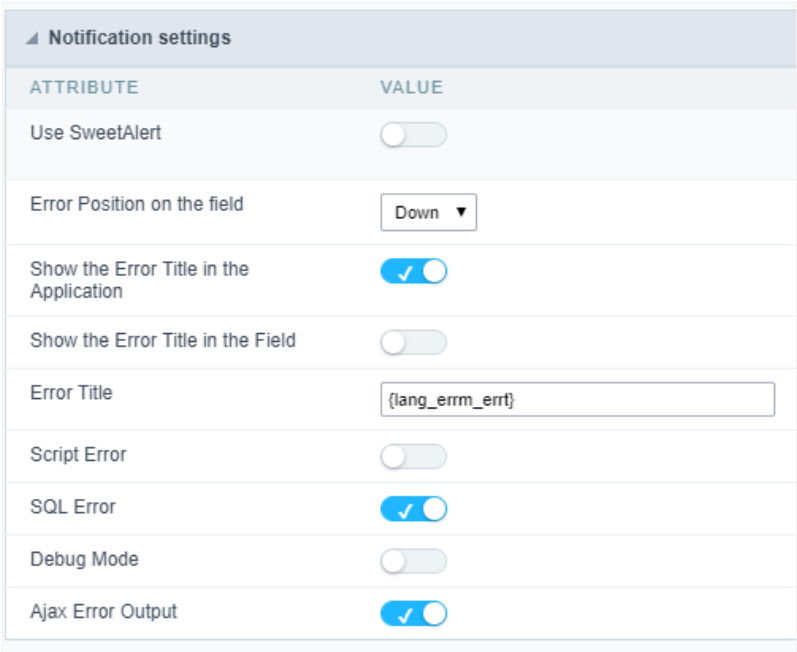
- **Attributes**
  - **Application Code** : It is the name that defines an application. An app can be renamed at the List of Application.
  - **Description** : This field contains a brief description of the application objectives.
  - **Documents Path** : The absolute path to store uploaded documents in the application.
  - **Image Directory** : The filesystem directory to store the application images.
  - **Application images** : Import images into the application to allows using them in the application.
  - **Language** : Set the default language of the application. Display all the application hints and messages in the selected language.
  - **Share Location Variable** : Define if the app shares the regional settings with other applications through a session variable.
  - **Charset** : Define a specific charset to use in the application.
  - **Share Theme Variable** : Define if the app shares the Theme settings with other applications through a session variable.
  - **Folder** : Define the project folder that contains the app.

- **Edit by Project** : Define if other project developers can edit the application.
- **Timeout** : Set the session runtime timeout in seconds. If the value is Zero, it assumes the default timeout of the PHP.
- **HelpCase Link** : It allows to associate a HelpCase file with the application.

- **Image Viewer**: Activates the image viewer features in the running application. By clicking on the image the user can drag, rotate, invert, and enlarge the selected image.



Notification Settings



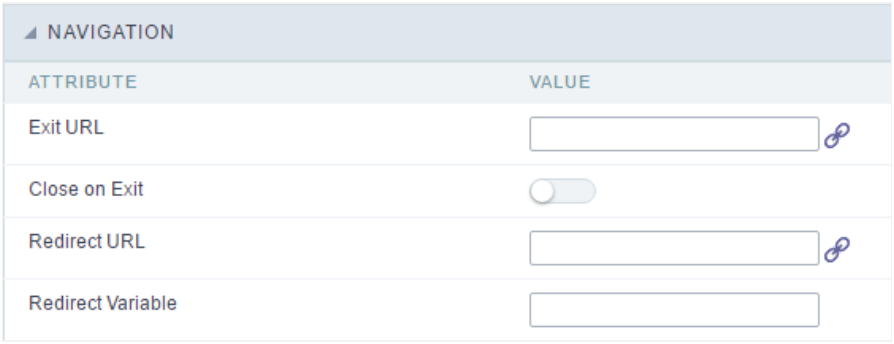
- **Use SweetAlert**: Use the SweetAlert to display messages from the application. When this option is active, it will replace the browser's "confirm" and "alert".
- **SweetAlert position using Toast** : The position to display error messages on the application.

- **Error Position on the field** : The position to display error messages when criticizing the field.
- **Show the Error Title in the Application** : Define to display the title line of the error message or not.
- **Show the Error Title in the Field** : Define to display the title line of the error message in the field or not.

- **Script Error** : Allows displaying the line code where there is an error..
- **SQL Error** : Allows displaying the SQL statement if it got an error.
- **Debug Mode** : Runs the application in Debug mode, showing all SQL statements the application is executing.
- **Ajax Error Output** : Enables the Ajax alert for debugging errors.

Navigation

This interface allows defining the navigating behavior of the application



Navigation Interface.

Exit URL

URL to where the user goes when he clicks on the “exit” button.

Close on Exit

Close the browser window when the user clicks on the “exit” button.

Redirect URL

Redirect to another URL in case there aren’t any global variables available.

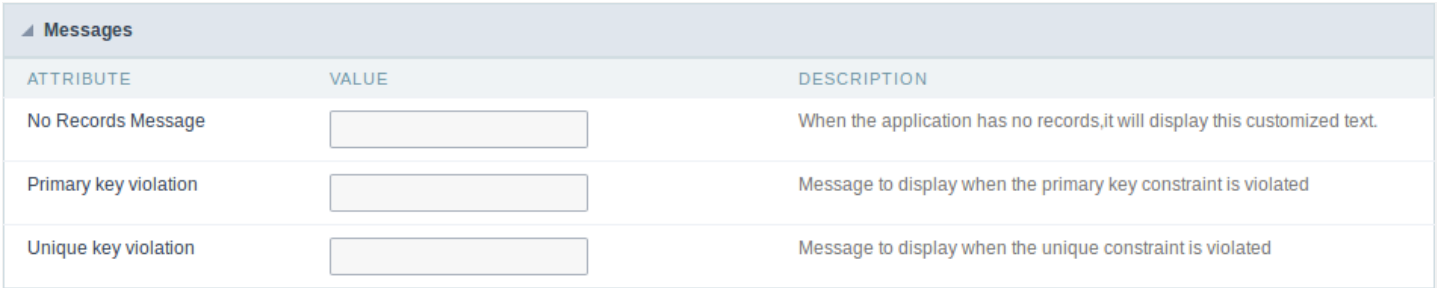
Redirect Variable

Creates a variable with the application name and sends it to the redirected application.

Messages

On this screen, you can redefine the default application messages to the end-user by customizing the validation messages for Insert, Update, and Delete. You can also define messages for SQL errors and the confirmation outputs.

Messages



No Records Message

The message displayed when the application has no records.

Primary key violation

The message displayed when there is a violation in the database's primary key constraint.

Unique key violation

The message displayed when there is a violation in the database's unique constraint.

Insertion Messages

Insertion messages

ATTRIBUTE	VALUE	DESCRIPTION
Message After Insert		Message to display after inserting a record.
Message to Confirm Insert		Message to display to confirm inserting of a record.

Message After Insert

The message displayed when inserting a new record.

Message to Confirm Insert

The message displayed if the end-user wants to confirm the inserting of a new record.

Update Messages

Update messages

ATTRIBUTE	VALUE	DESCRIPTION
Message After Update		Message to display after updating a record.
Message to Confirm Update		Message to display on the update confirmation

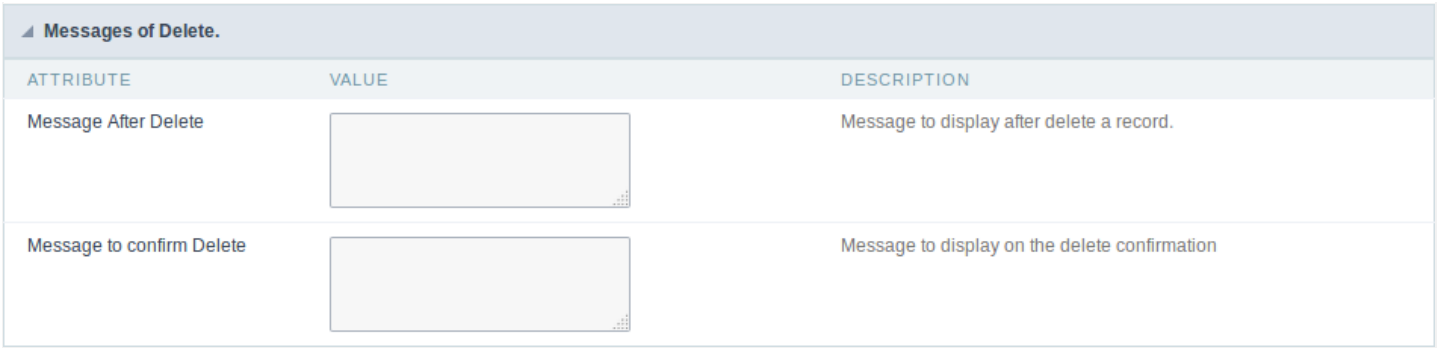
Message After Update

The message displayed when updating a record.

Message to Confirm Update

The message displayed if the end-user wants to confirm the changes of a record.

Messages of delete



Message After Delete

The message displayed when deleting a record.

Message to confirm Delete

Displays a customized message asking to confirm the record deletion.

If you make no changes, the application will use the default values for the messages. Those values can be configured in Locales -> Application Language.

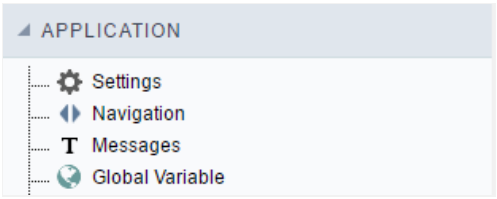
Global Variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target="_blank"} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



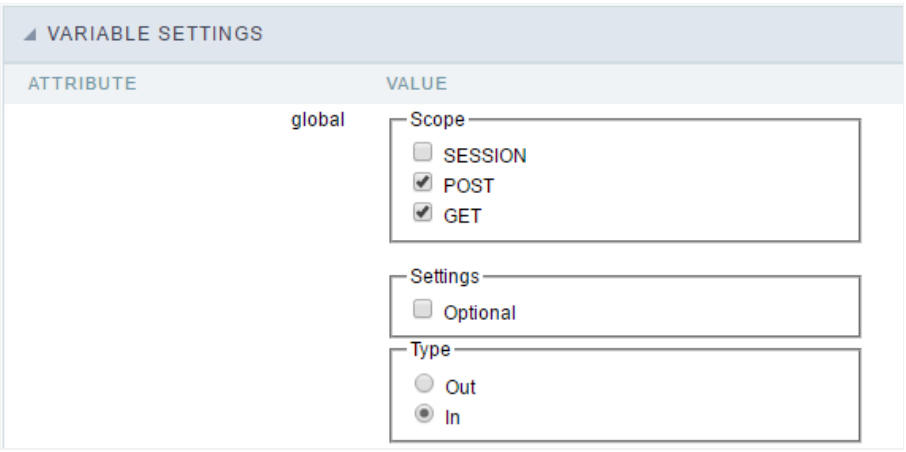
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

Indicates where the application is using the variable.

Synchronize table

This process performs a comparison between the definitions of application fields and fields of the connected database table. If there is any change within the connected table, such as adding a new field, exclusion, or a change in the data type, the changes will appear visible as in the image below:

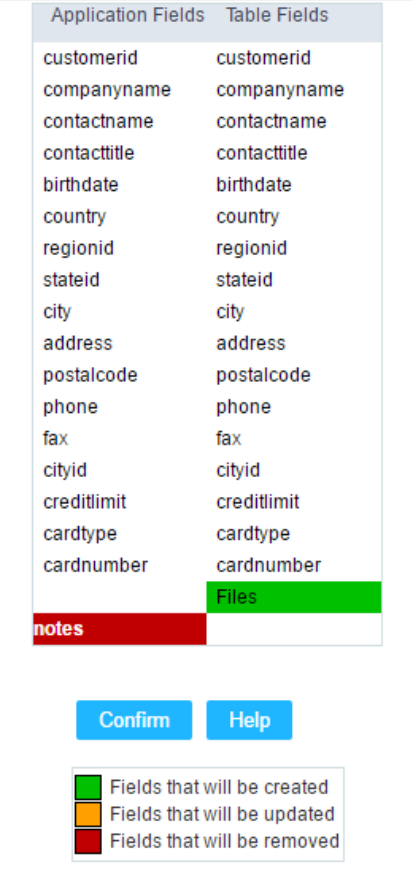


Table synchronization interface.

After accessing the function from the link “Synchronize table” you will see a comparison table between the application fields (left table) and the database table fields (right table), above image, the fields highlighted in “red” will be deleted of the application, the fields highlighted in “Green” will be inserted in the form, and the fields highlighted in “orange” will be updated, so data type will be updated.

To rename a field in the database table, the table synchronization effect, by comparison, will be the same as deleting a field that existed in the application and the inclusion of a new field.

Below you can see a video showing the process:

Form Links Overview

This feature allows the developer to create links between applications of the same project, expanding the integration of applications. All link options are grouped under the Application Links menu.

Application Links

In the first access to the menu, we can see the list of existing connections in the application.

> If the application does not have a configured link, the application list screen will be displayed with the message: **This application does not have any link. Click here to create one now.**

ID

Link identification ID.

Type

The type of link created, some links such as Edit Link allow only one link per application. In this case, the developer will be able to check the connection types that already exist in the application

Target Application

This info show the target application name.

Actions

This column has edit options for the links.

Properties

It allows accessing the binding properties where it is possible to configure the binding behavior.

Link

Displays the links screen, where it is possible to configure the link that was made with the application informed in the target application column. In this option it is possible to change the parameter passed in the connection as well as the target application.

Delete

Permanently deletes the connection in the applications.

> Through the option Restore Applications, it is possible to get a previous version of the application, making it possible to recover the link deleted.

Links Type

The Form application has the following links options

- [Application Link][ligacao_edicao] - Allows you to link a form with any project application.
- Capture Link: It allows the creation of links from the filter fields of the Grid application, in order to enable the recovery of the value to fill in the field, with another Grid application of the project.
- Field Link: Allows you to create a link through the application fields to any application in the project.
- Button Link: It allows creating a link through the buttons created by the developer in the query application, to any application in the project.

Form Applications Link Types x Target Application available to link

Check the links types of the Form application and the target applications available for each link types.

Only **Capture link** have some restrictions in the target application use, allowing only to use of Grid application.

	Application Link	Capture Link	Button Link
Grid			
Chart			
Form			
Control			
Search			
Menu			
Tree Menu			
Tabs			
PDF Report			
Dashboard			
Blank			
Calendar			

Related Links

- Master/Detail linked to a Grid
- Capture information from a grid
- Grid and Form in the same page
- Master/Detail
- Form overview
- Macro sc_master_value
- Linking applications
- Macro sc_link

Related Videos

Application link

Este tipo de enlace permite crear un enlace desde un Formulario hacia cualquier tipo de aplicación.

En las opciones de tipos de enlace, seleccionaremos la Ligación de Aplicación. Al elegir esta opción, será posible acceder a otra aplicación desde el formulario.

![Interfaz de creación de enlace de aplicación.][ligacao_criando_aplicacao]

Application List

All project applications available for the link will be listed. In this step, the developer must select the destination application that will be used in the connection.

![Tela com a lista de aplicações disponíveis para ligação de edição][img_lista_apps]

Finding Applications

To assist in locating the application, you can use the **grouping buttons** or the **Search** field.

Search field

The search is performed by application name.

Grouping Buttons

In this grouping there are three visualization options:

All

Lists all applications in the project alphabetically **a-z**.

Example of applications without clustering

By folder

Lists applications grouped by their folder, according to the organization of applications in the Explorer Project.

Example of applications grouped by Folder

- **Item 1** - List of folders created in Explorer Project. Select the folder and all applications in the folder will be displayed on the **item 2**.
- **Item 2** - List of applications in the folder selected in the__Item 1__.

By type

Lists applications grouped by their respective type.

Example of applications grouped by type

Item 1 - Types of existing applications in Scriptcase. Select the type of application you want to list, the applications corresponding to the selected type will be listed in **item 2**. **Item 2** - List of applications referring to the type selected in **item 1**.

When selecting the destination application, click on **Next »**.

Parameter Definition

In this step, the developer must inform the values that will be passed to the parameters of the destination application.

These parameters can be:

- Global Variables - Which can be defined in the events or in the target application's SQL.
- Primary Key - For connections created with applications of the types: Form and Calendar.

For the **Edit Link** to function correctly, it is necessary for the primary key field of the form to receive a value, which can be defined by a global variable, a field from the grid, or by using the Fixed Value option.

Parameters

Lists all parameters defined in the target application.

Type

Defines the source type of the value that will be sent to the parameter defined in the destination application.

The options are:

- **Fields** - It uses the value of a field from the source application as a parameter.
- **Variable** - It uses the value of a global variable, defined in the source application, as a parameter. **This option will only be listed if a global variable is defined in an event in the source application**.
- **Fixed value** - It uses a fixed value, defined in the value column, as a parameter. **In this option only alphanumeric values are allowed**.
- **No value** - Using this option no value is passed. When using this option as a parameter for a primary key, the target application will be displayed in include mode.

Value

Defines the value that will be sent as a parameter. The options in this column change according to the selected **Type**.

- When selecting **Campo** - The value column will list all the fields of the source application, which will send the value.
- When selecting **Variable** - All global variables defined in the source application will be listed.
- When selecting **Fixed value** - A field will be displayed for the value to be informed. The use of variables is not allowed in this option and we must use only alphanumeric values.
- When selecting **No Value** - In this case, no value will be sent to the parameter.

Refresh Button

This button enables ajax reloading of the list of parameters, in this way it is possible to change or add a parameter in the destination application without the need to restart the creation of the connection in progress.

Properties

The **Properties** screen allows you to configure the redirection behavior between applications in Scriptcase.

Link Properties

In this screen, we will configure the link properties.

Redirect after insert:

This setting provides three options that will apply after an insertion.
Yes: Selecting this option, the application will go to the linked application as soon as a new record is inserted.
No: Selecting this option, the application will not go to the linked application as soon as a new record is inserted.
Exit Application: Selecting this option, the application will go to the exit application or to the application that initially called the form, after the record insertion.

Redirect after update:

This setting provides three options that will apply after an update.
Yes: Selecting this option, the application will go to the linked application as soon as a record is edited.
No: Selecting this option, the application will not go to the linked application as soon as a record is edited.
Exit Application: Selecting this option, the application will go to the exit application or to the application that initially called the form, after updating the record.

Form Properties

These properties are only available in links where the destination application is a form, this configuration block defines whether or not to display the **Update** (*Insert*, *Update* and *Delete*) and **Navigation** (*first*, * previous*, *next* and *last*) of the target form, according to the toolbar settings of the called application.

It is important to point out that the attributes in the Form Properties block do not override the toolbar settings of the called form, they only determine whether the buttons configured there will be displayed or not in the link.

> This attribute is not available if the **Link Operation Mode** attribute is set to *Open in an iframe*. In this configuration, the form's buttons will be displayed on the toolbar of the link's source application. Click here for more details on the query toolbar.

Enable Insert Button

Sets whether or not to display the **Insert** button on the target form. For this, it is necessary that the button *include* is selected to be displayed in the settings of the toolbar of the appplication.

Enable Update Button

Sets whether or not to display the **Refresh** button on the target form. For this, it is necessary that the button *change* is selected to be displayed in the settings of the toolbar of the appplication.

Enable Delete Button

Sets whether or not to display the **Delete** button on the target form. For this, it is necessary that the button *Delete* is selected to be displayed in the settings of the toolbar of the appplication.

Enable Navigation Buttons

Determines whether or not to display the **Navigation Buttons** (*First*, *Back*, *Page Navigation*, *Next*, *Last*) configured in toolbar of the target form.

Keep the WHERE clause in the target application

> This attribute is only available when enabling navigation buttons.

This attribute makes it possible for the WHERE clause of the source application (Query) to be maintained in the destination application. This means that the filters performed in the query will be inherited by the target form, enabling better quality in the records displayed for editing.

Related Links

Related Videos ▷

- Master/Detail linked to a Grid
- Capture information from a grid
- Grid and Form in the same page
- Master/Detail
- Form overview
- Macro sc_master_value
- Linking applications
- Macro sc_link

Form Field Link Settings

Creating a field link

Allows to create a link, represented through a link, from a grid’s column to any project’s existing application. All the links are displayed in a dropdown if there are more than one link to the same field.

In the types of link option, we will choose the **Field link**.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project’s applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

![Same window][mesma_janela]

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open the application in a tab inside the menu:

Choosing the option **Yes**, the application will be open in a **menu tab**, when executed from a menu application, instead of opened in another browser window.

Open in an iframe

When we use this option the target application will be displayed in the same window of our application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Iframe properties

In this settings, we can define some iframe details that will display the target application.

Iframe position in relation to the main application:

In this option, we will define in which position in relation to the main application the iframe will be displayed, there are four options:

Below: The iframe will be displayed below the main application.
Above: The iframe will be displayed above the main application.
Right: The iframe will be displayed on the right of the main application.
Left: The iframe will be displayed on the left the main application.

Iframe height:

Allows to set the iframe height.

Iframe width:

Allows to set the iframe width.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Grid’s Properties

In those settings we can define how our target grid will be displayed. We inicially have six options, that are:

![Form proprieties][propriedades_consulta]

Initial module:

We can define how our application will be executed in those two options:

Grid: The application will be executed as a grid.
Search: The application will be executed as a search.

Quantity of rows:

In this option we can define the quantity of rows that will be displayed in the target application.

Quantity of Columns:

In this option we can define the quantity of columns that will be displayed in the target application.

Pagination:

In this option we can define if the target application’s pagination will be total or partial, in case partial is the selected option the quantity of rows will be the informed previously.

Enable header:

In this option we can define if the target application’s header will be displayed.

Enable Navigation buttons:

In this option we can define if the buttons **first, previous, next and last** will be available in the target application.

Related Links

- Master/Detail linked to a Grid
- Capture information from a grid
- Grid and Form in the same page
- Master/Detail
- Form overview
- Macro sc_master_value
- Linking applications
- Macro sc_link

Related Videos

Form Capture link Settings

Creating a Capture Link

The capture link is used to return a value from a **Grid** to a **Form** field.

In the types of links options, we will choose the **Capture Link**. After selection this option, we should also choose which field we want to return the value of.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

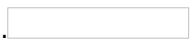
We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side. 

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Application with no parameters

However, when the target application does not have any defined parameters, the following screen is displayed:

Clicking in the button, you will be taken to the target application to create a parameter, so you can use the update button in the **parameters definitions** to refresh them.

Link properties

In this screen we will set the application display mode that will be called in the link.

In this type of link there is only one display option:

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Modal:

In this option we are going to define if the target application will be opened in a Modal.

Yes: This option will make the target application be opened in a modal. **No**: This option will make the target application be opened in a new window.

If **Yes** is selected in the previous option, the Modal **Height** and **Width** will be available.

![Modal with yes][modal2]

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

If **No** is selected in the previous option, only those options will be available.

Allows to modify manually in the update:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Allows to modify manually in the insert:

Indicates which shortcut key to the button add new register.

Allows to modify automatically in the update:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Allows to modify automatically in the insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Related Links

- Master/Detail linked to a Grid
- rmation from a grid
- m in the same page
- Master/Detail
- Form overview

Related Videos

- Macro sc_master_value
- Linking applications
- Macro sc_link

Form Button link Settings

Creating a Button Link

Allows the developer to create a link where the call to the other application will be done through a button.

In the type of links options, we will choose the **Button Link**. Choosing this option it will be possible to create a link to any other application.

List of applications

After selecting this option, The list of applications to what you want to create a link will be displayed.

This screen can be viewed in the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open the application in a tab inside the menu:

Choosing the option **Yes**, the application will be open in a **menu tab**, when executed from a menu application, instead of opened in another browser window.

Choosing the option **Yes**, there are also some other settings to be displayed:

Title to the tab:

This option allows the developer to set a title to the tab that will be opened when used in a Menu application.

Hint to the tab:

This option allows the developer to set a message to be displayed when the mouse cursor is over the tab Menu.

Active tab icon:

This option allows the developer to set an icon to be displayed in the tab when used in a menu application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Height:

Allows to set the Modal's height.

Width:

Allows to set the Modal's width.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Related Links

- Master/Detail linked to a Grid
- rmation from a grid
- m in the same page
- Master/Detail
- Form overview
- Macro sc_master_value
- Linking applications
- Macro sc_link

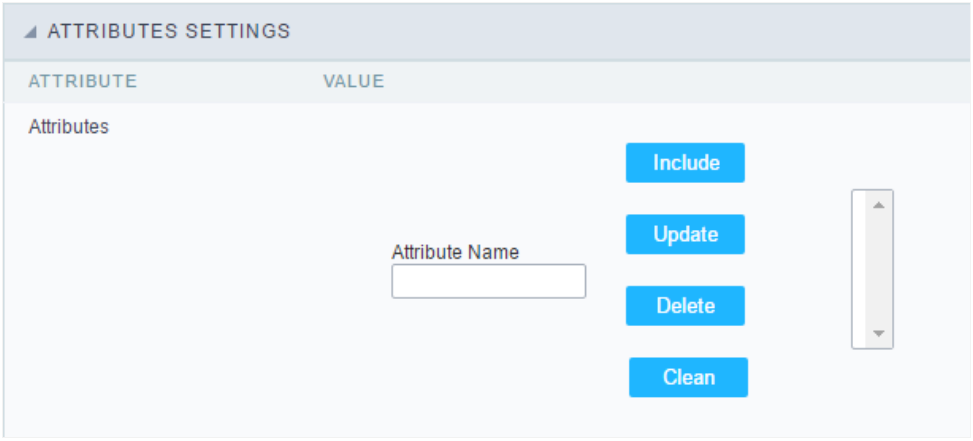
Related Videos

Form Programming

ScriptCase has incorporated the concept of Object Oriented programming, using attributes, resources, methods and libraries. It is possible to create your own business rules in applications, and by using these concepts you can reap huge rewards in terms of better organization and improved development.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

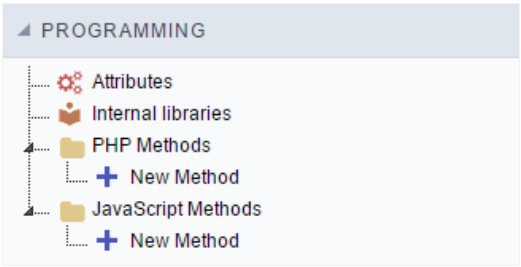
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

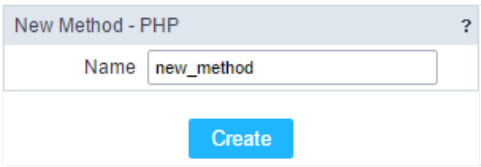
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

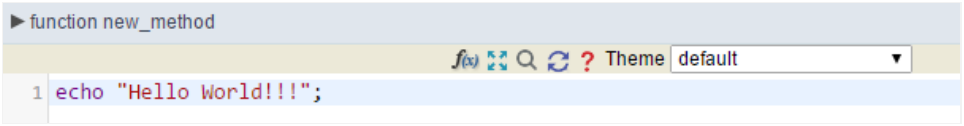


- Give the method a name and click Create. As in the image below.

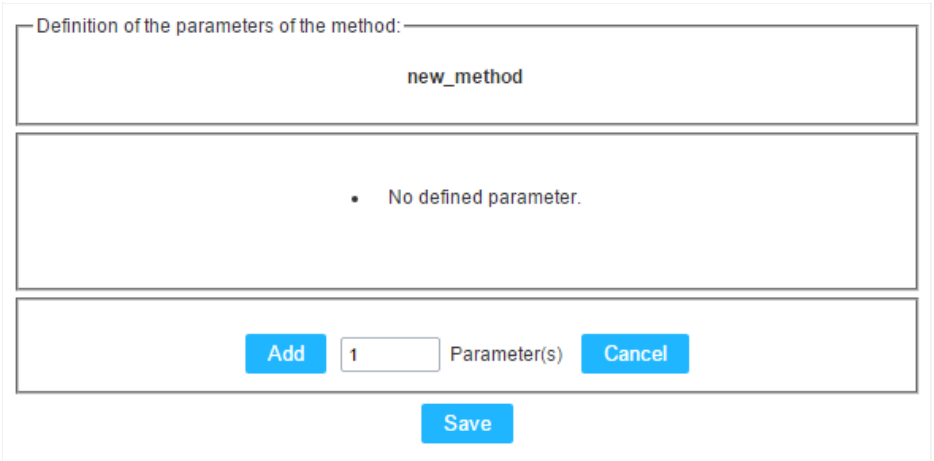
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



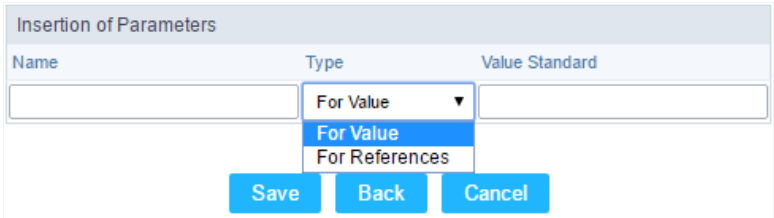
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

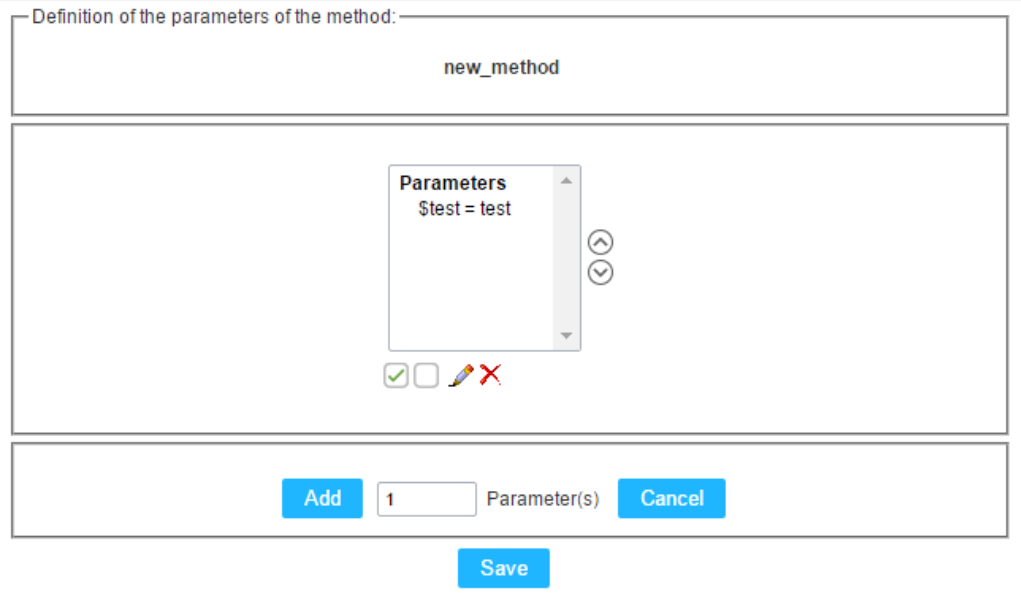


- **Name** : Type in the variable's name.
- **Type** : Selecting the type of variables: For Value or For Reference.

- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:

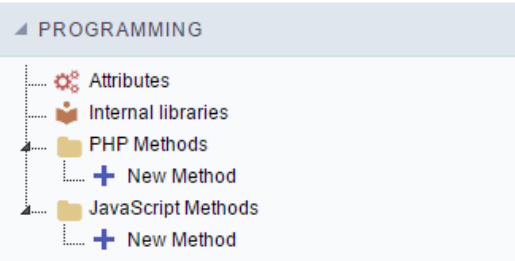


- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

JavaScript Methods

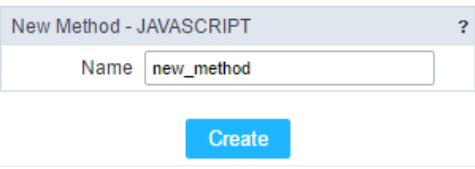
Methods are function or procedures declared by the developer, that helps when appling the routines. Using methods in the applications allows to reuse of the your code throughout the application, optimizing the development experience.

Creating a new method



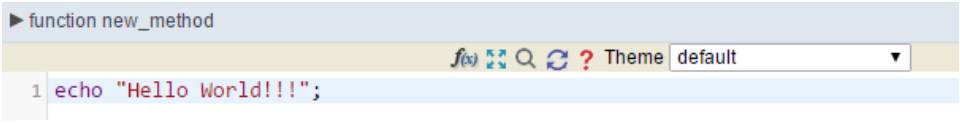
JavaScript method creation Interface

- Define a name for the method and click on Create. Like the image below.

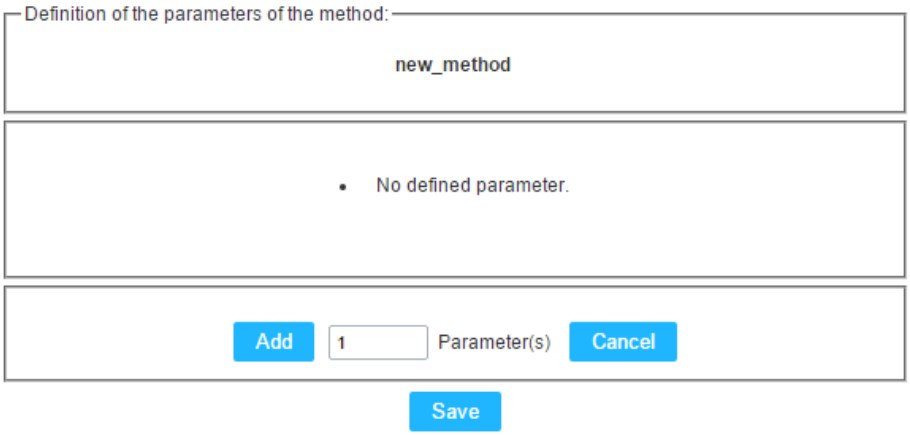


Include Method.

- Methods can receive parameters.

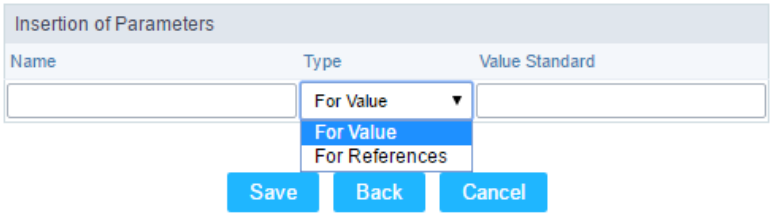


  - Add the amount of variables:



JavaScript method parameter definition

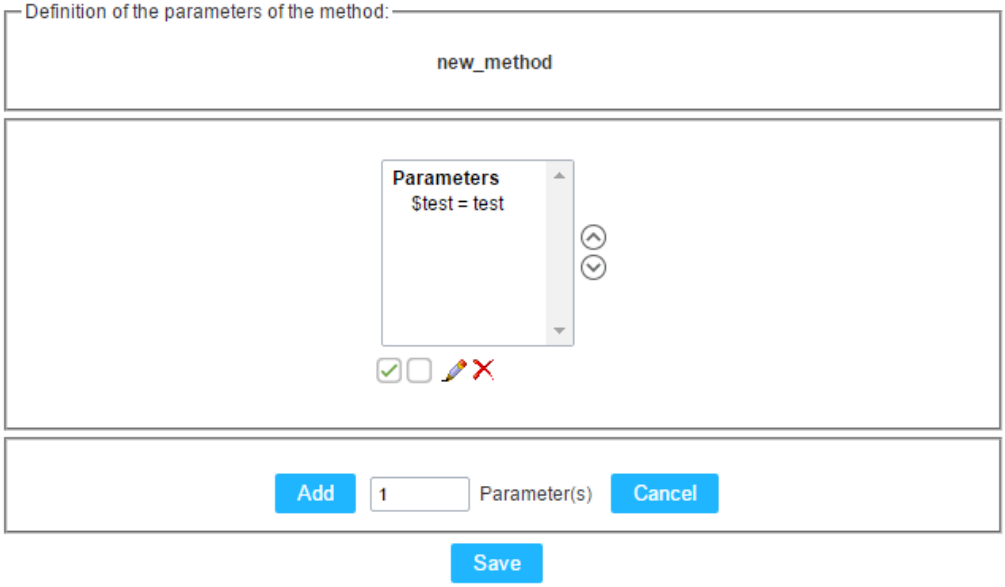
  - Defining the variables:



Defining the variables

    - **Name** : Type in the variable's name.
    - **Type** : Selecting the type of variables: For Value or For Reference.
    - **Value Standard** : The parameter's value used to initialize when calling the method.
      References allows to create a second name for a variable that you can use to read and modify the original variable's information.

  - Editing a parameter:



    - : Checking all the parameters in the parameter list.
    - : Uncheck all the parameters in the parameter list.
    - : Edit the selected parameter of the list.
    - : Deletes the selected variable of the list.

Related Links

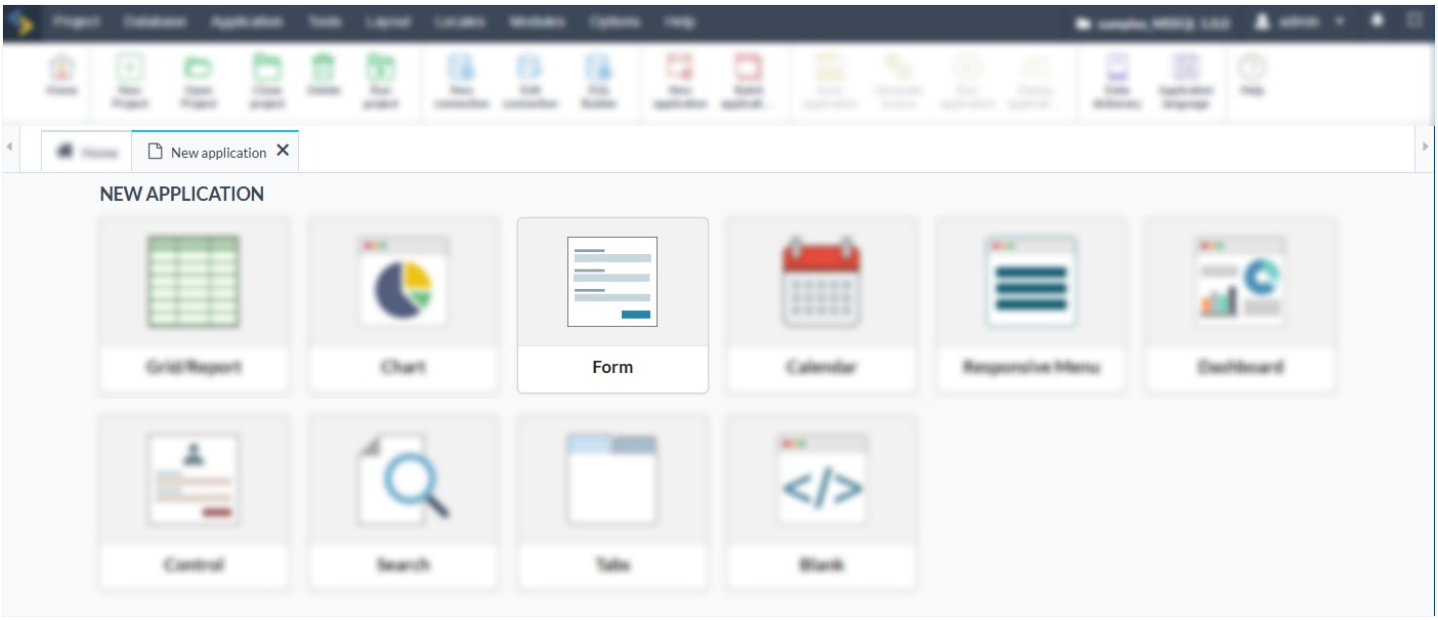
Related Videos

Creating a Form Application

New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.



Application Data

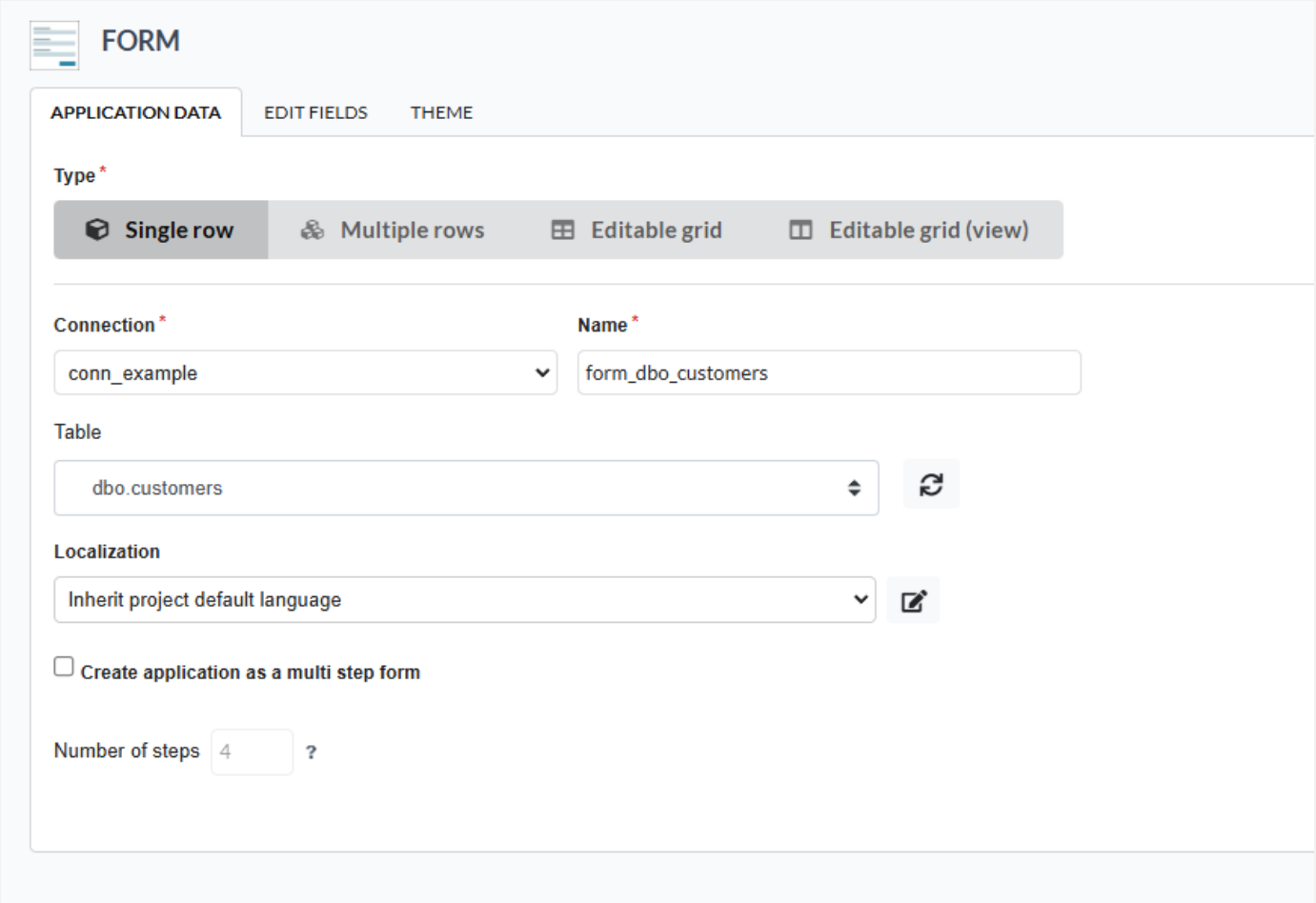
When selecting the application, the following screen will appear.

On this tab, you must select the connection to be used in the application. Once the connection is selected, the tables will be loaded and displayed in the **Table** field.

You can select more than one table when creating applications.

After selecting the tables and fields to be part of the application, the application name will be generated using the type + table name.

Example: form_orders
However, you can change the application name to any other name you prefer.



Connection

Defines the connection used for the creation of the application. The existing connections in the project will be listed.

Name

The name of the application being created. It cannot contain special characters.

Table

Defines the tables to be used in the application. (Form and Calendar can only use one table).

Localization

The language of the application to be created. The project's default language is automatically selected.

Create application as a multi step form

When this option is selected, the form will be configured in a step format, dividing the filling process into multiple pages.

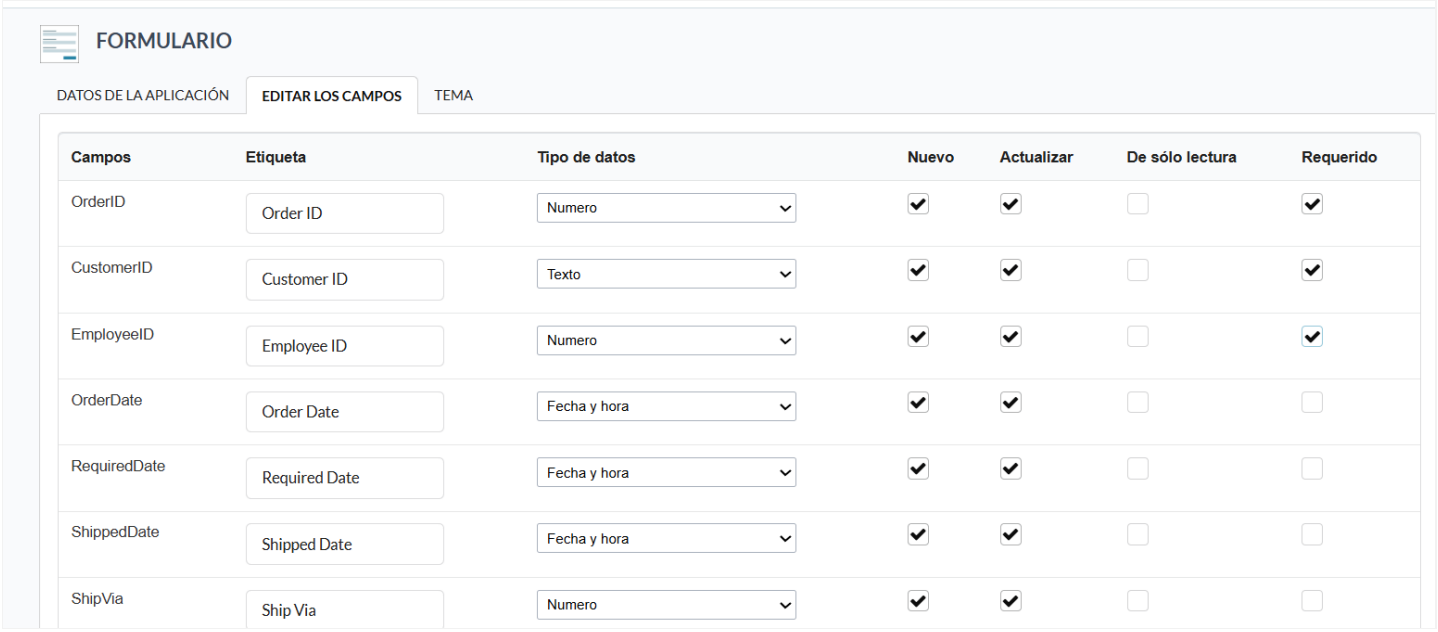
This functionality is available only for single-record forms. After creation, it is possible to revert to the traditional format by adjusting the page settings.

Number of steps

Defines the initial number of steps the multi-step form will have, creating the blocks and pages that compose the form automatically. The fields will be distributed across the created pages and can be rearranged later by adjusting the field positioning.

Edit Fields

This screen displays the fields of the selected tables and allows adjustments to be made before creating the application, such as changing the data type, display name, and other configurations.



Fields

Names of the database fields.

Label

Names of the fields in the generated application's interface.

Datatype

Specifies the field's data type.

New

Defines the fields available for insertion.

Update

Defines the fields available for the Update.

Read-Only

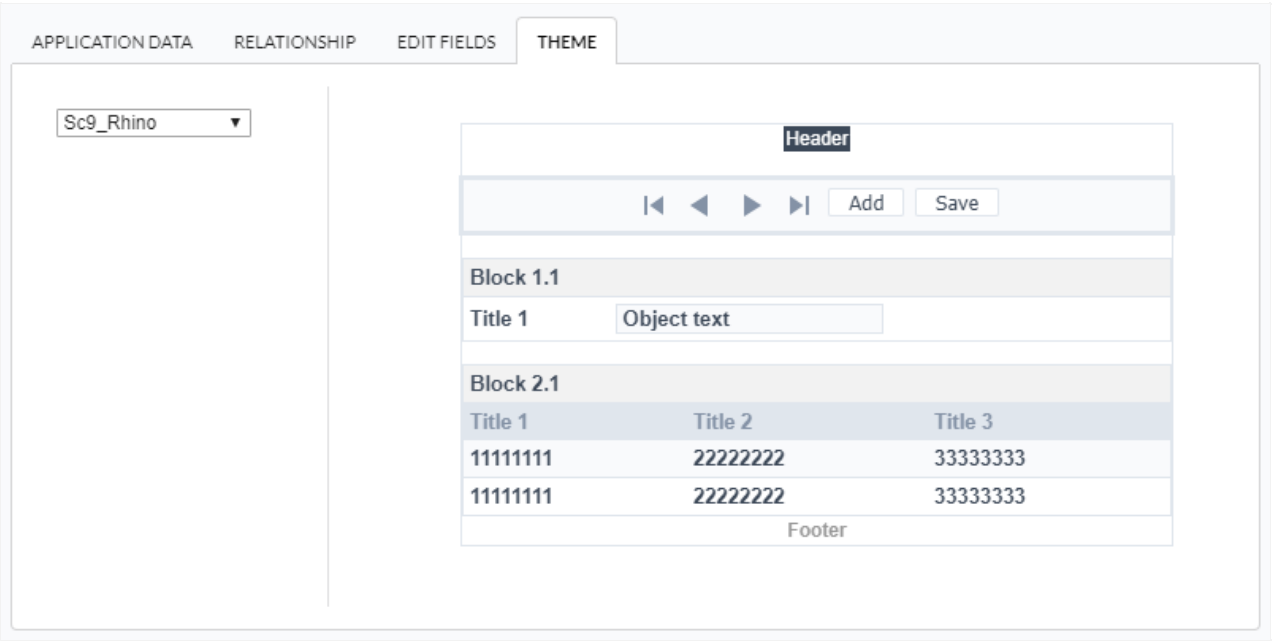
Defines whether the field will be read-only.

Required

Defines whether the field will be mandatory.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.



Related Link 🔗

Related Videos ▷

Form Mobile Optimization

The Mobile settings allows to define the automatic optimization of generated applications to run on mobile devices.

See below the available settings.

Mobile optimization

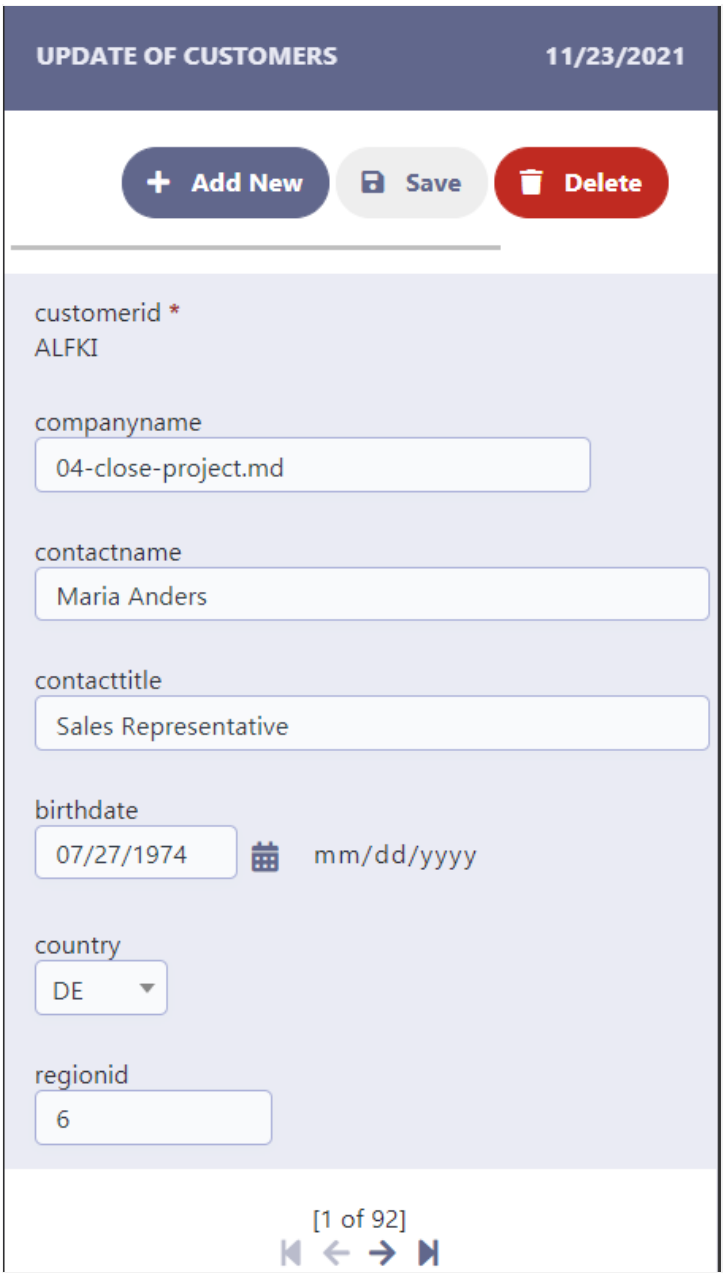
Enable mobile optimization

This option changes how the application HTML elements works, adapting them automatically to run on mobile devices.

It enables this option by default when creating a new application.

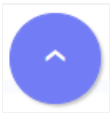
See some examples of adapted screen:

Single registration form mobile optimization example.



Enable back to top button

It enables the displaying of the button **back to top button** in the Grid, Filter, Detail and Summary modules.



It enables this option by default when creating a new application.

Back to top button position

Defines the position of the **back to the top** button:

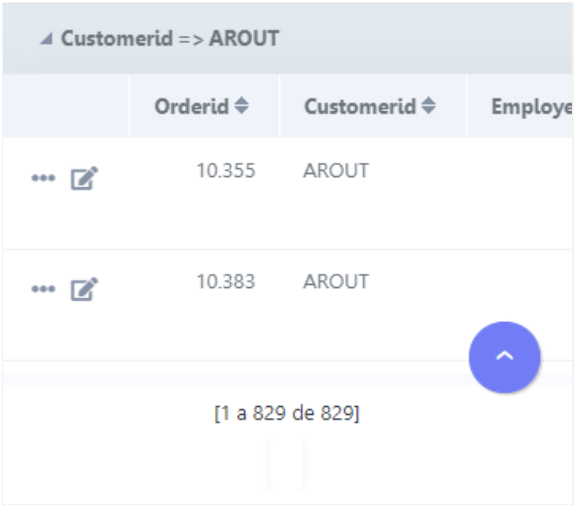
The options are:

- **Right** - Position the button in the bottom right corner;
- **Left** - Position the button in the bottom left corner;

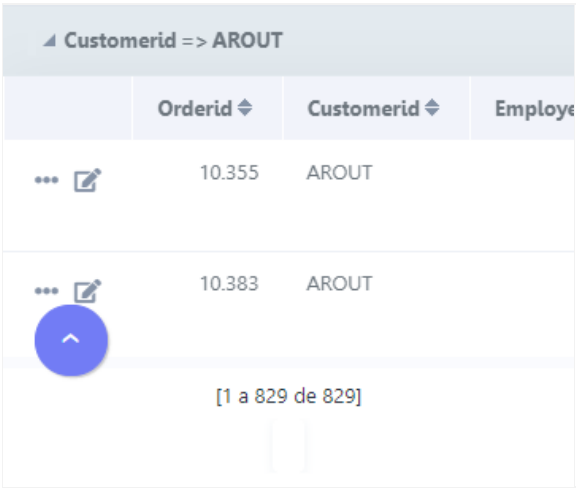
When creating a new application, the button is configured on the right by default.

See below for placement examples.

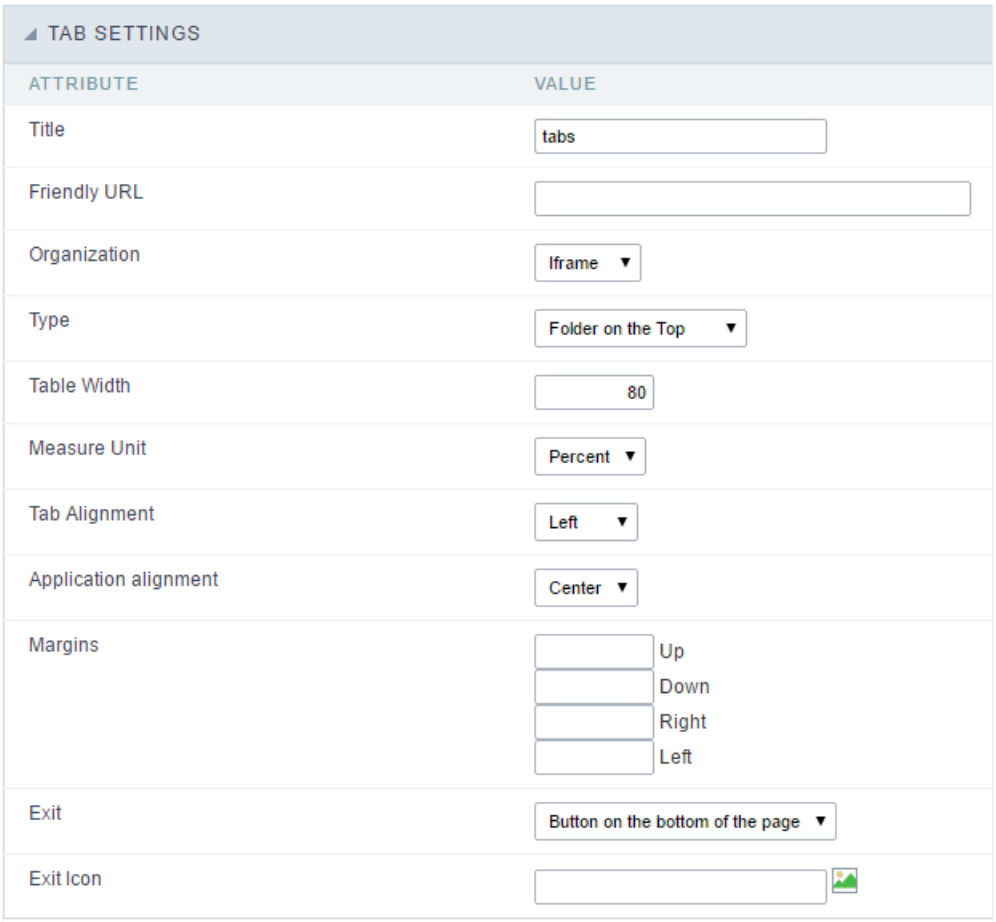
Button in the bottom right corner



Button in the bottom left corner



Tab Settings



Tab settings interface.

Within the Tab settings you can configure the following attributes:

- **Title** : Application title that will be displayed in the toolbar.
- **Friendly URL** : This field allows you to customize the URL that will be called by the application, the allowed characters are the same available on regular URLs: a-z, A-Z, 0-9, -_. This option can also be changed within Scriptcase Home interface, on the “Friendly URL” column from the applications grid.
- **Organization** : Allows the applications to run inside folders via iframe or embedded on the tabs application.
- **Type** : This option sets how the folders will be organized, it can be:
  - **Folder on the Top** : Displays the folders above the application.
  - **Folder on the Bottom** : Displays the folders below the application, next to the footer.
  - **Folder on the Left** : This option allows the folders to be aligned in the left, like a vertical menu.
  - **Folder on the Right** : This option allows the folders to be aligned in the right, like a vertical menu.
  - **Menu on the Top** : This option is similar to “Folder on the Top”, but its visual schema has the menu style.
  - **Menu on the Bottom** : This option is similar to “Folder on the Bottom”, but its visual schema has the menu style. .
  - **Menu on the Left** : This option is similar to “Folder on the Left”, but its visual schema has the menu style.
  - **Menu on the Right** : This option is similar to “Folder on the Right”, but its visual schema has the menu style.
  - **Side by Side** : Display all applications in folders, one next to the other.
- **Table Width** : Main table size.
- **Measure Unit** : The unit used by table size. It can be: Pixel or percentage..
- **Tab Alignment** : This option sets the tab alignment position. It can be: center, left or right.
- **Application alignment** : This option sets the application alignment position, It can be: center, left, or right.
- **Exit** : This option allows you to define how the exit will be presented in the application, the options:
  - Button on the bottom of the page;
  - Folder;
  - Button in the toolbar

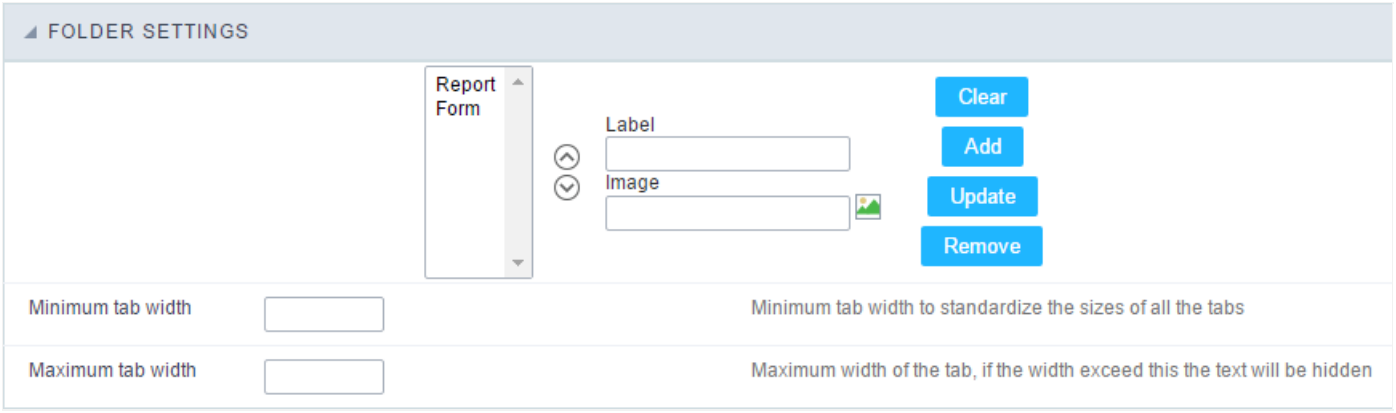
    - In view.
- **Exit Icon** : This option allow you to select an icon for the “Exit” option in the tab. Use the image manage to select or import a customized image.

Related Links 🔗

Related Videos ▷

Folder settings

You will use the folder settings options to manage each folder link to an existing application.



Folder Settings Interface.

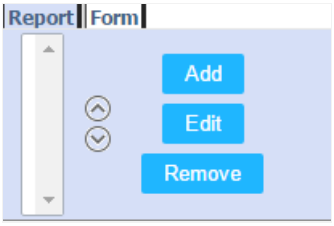
- **Add a folder** : To create a tab, click in 'Add' and fill these fields: * **Label** : Application name that will be displayed in the tab or menu (examples: Personal Data, Finance Data, etc.); * **Image** : Choose the image that will be used for this tab;
- **Update** : Double click on the folder name (left column / list of folders) and then update the data. Finally, click on the 'update' button.
- **Clean** : Double click on the folder name (left column / list of folders) and then delete the item.
- **Remove** : This button when clicked, clears all existing folders in the box (the box located in the left).

Related Links

Related Videos

Application settings

In this folder you can set which applications will be part of the tabs implemented, as follows:



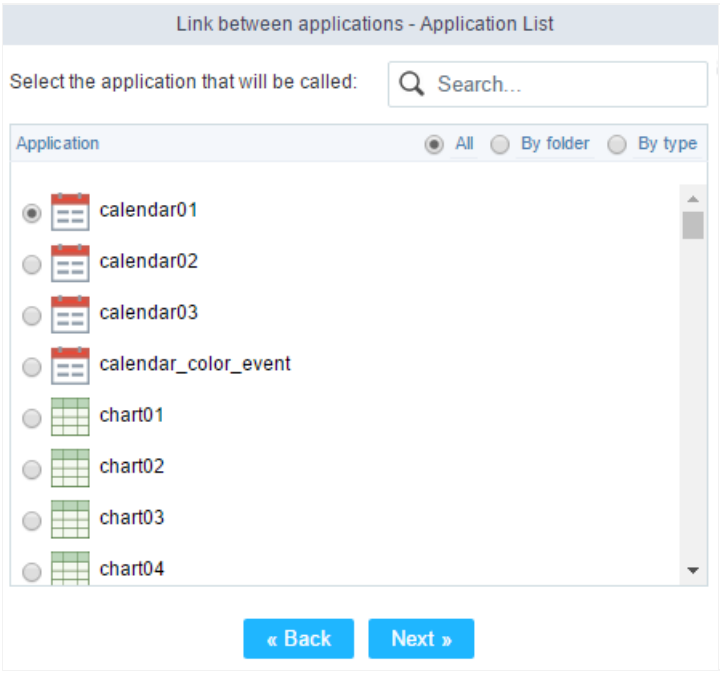
Application Settings Interface

Initially you must click on the tab that you need to update. You have the options:

- **Add** : Click “Add” button, and then the following screen will be displayed for selection of applications.
- **Edit** : Select the application for the link and then click the “Edit” button. The following steps are the application selection and setting the size of the iframe, as requested in the process to add applications.
- **Remove** : To remove just select the application you want to delete in the folder, and then click “Remove”.

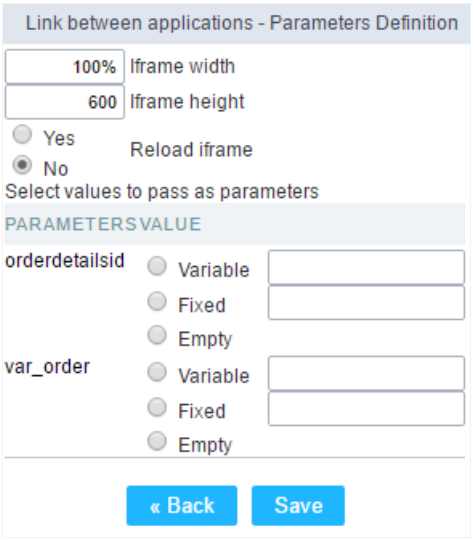
Adding and editing applications

Select the application to be placed in the folder, and click the continue button, you must see the screen below:



Application Link Interface.

Then the following screen will be displayed to complement with the link parameters to the selected application:



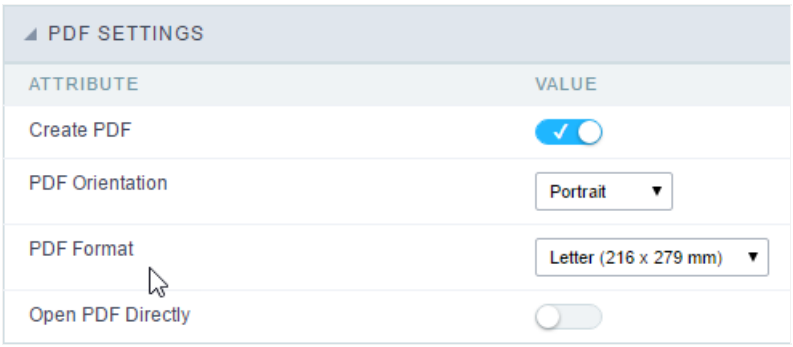
Parameters set Interface..

- **Iframe width** : Width set to be used inside the folder to display the application window, leaving 100% the application will be adjusted more easily.
- **Iframe height** : Height set to be used in the iframe to display the application window, if the it is too small, the application will create a scroll bar to the window.
- **Reload Iframe** : When a folder is selected the applications contained in the same run and the result is displayed, with this option turned on “Yes” when you click on the folder that contains the application cache is created with the page loaded, this cache will be used if the user browse for other folders and return to the folder that has the application with this option enabled.
- **Parameters / Value** : For each parameter of the application we are putting in the folder, there are three options to choose:
  - **Variable** : Global variable name where the content will be retrieved.
  - **Value** : Content to be loaded into the parameter at run time.
  - **Empty** : For this parameter the content will not be the responsibility of the Tab application

Related Links 🔗

Related Videos ▷

Tab Application Export PDF Settings



PDF Settings Interface.

- **Create PDF** : This option allows to define if the application will have output in PDF format.
- **PDF Orientation** : This option sets the direction of the reports generated in PDF, it can be: portrait or landscape.
- **PDF Format** : Pages format of the reports generated in PDF, such as Letter, A4, A5, etc.
- **Open PDF Directly** : This option indicates whether the application will allow us to generate the PDF directly via button or through a link to the PDF.

Related Links 🔗

Related Videos ▷

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application.

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Related Link

- Security system template

Related Videos

- Form overview
- Security Module (fundamentals)
- Security Module (learning SC from scratch)

- Log module (fundamentals)
- Log module (learning SC from scratch)
- Login Screen
- Control Module (macro sc_log_add)

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Related Link

- Security system template

Related Videos

- Form overview
- Security Module (fundamentals)
- Security Module (learning SC from scratch)
- Log module (fundamentals)
- Log module (learning SC from scratch)
- Login Screen
- Control Module (macro sc_log_add)

Headers

Desabilitar o Auditor XSS

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Related Link

- Security system template

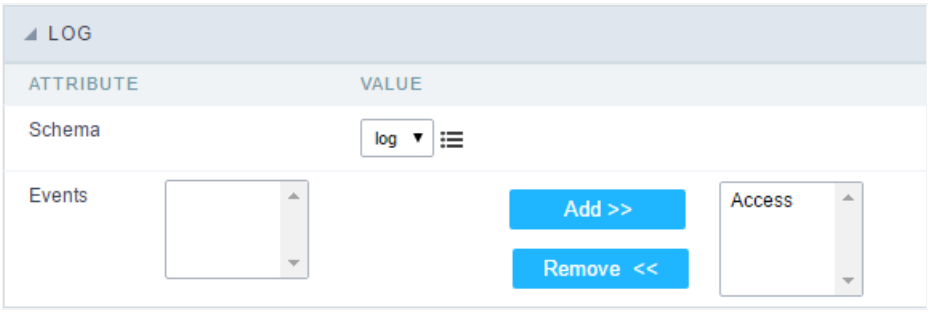
Related Videos

- Form overview
- Security Module (fundamentals)
- Security Module (learning SC from scratch)
- Log module (fundamentals)
- Log module (learning SC from scratch)
- Login Screen
- Control Module (macro sc_log_add)

Tab Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Related Link

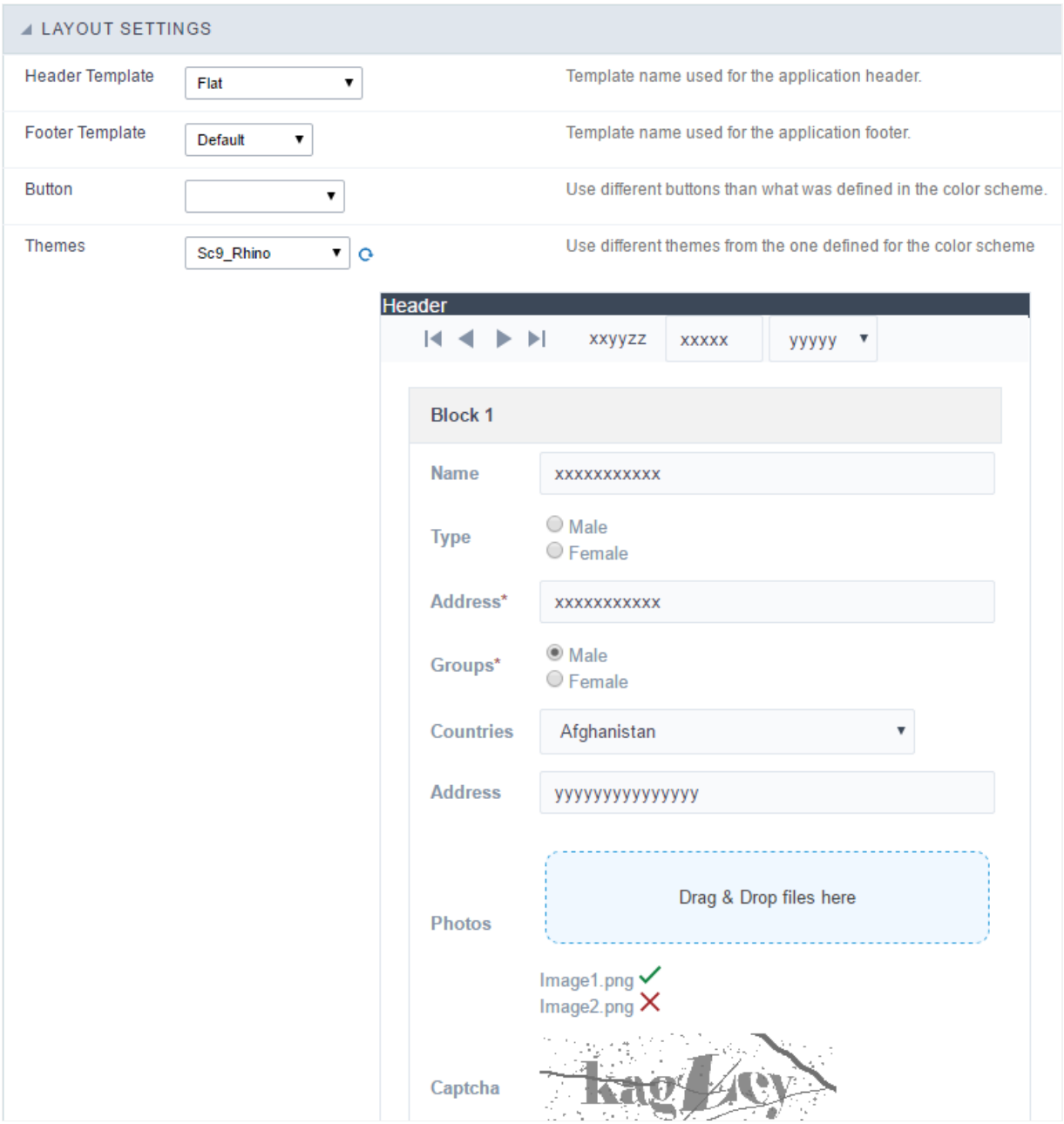
Related Videos

Tab Application Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout Settings

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

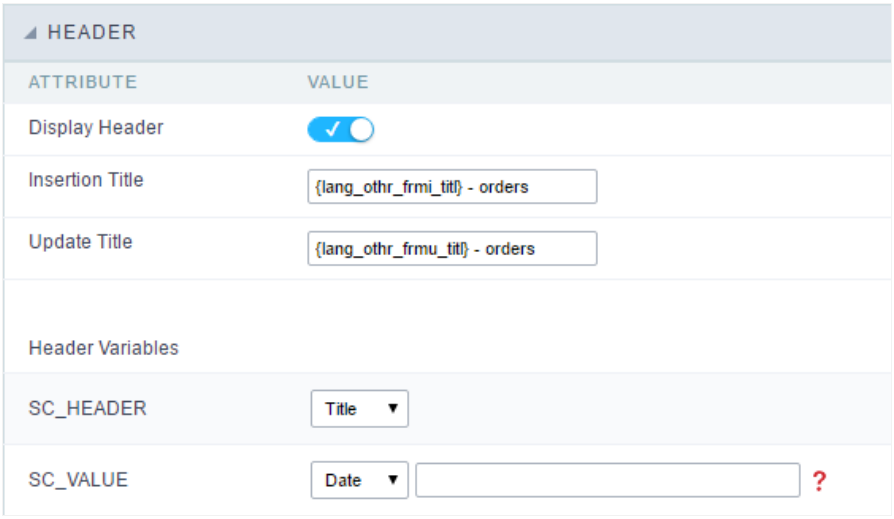
Allows choosing the button theme for the application.

Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**".
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

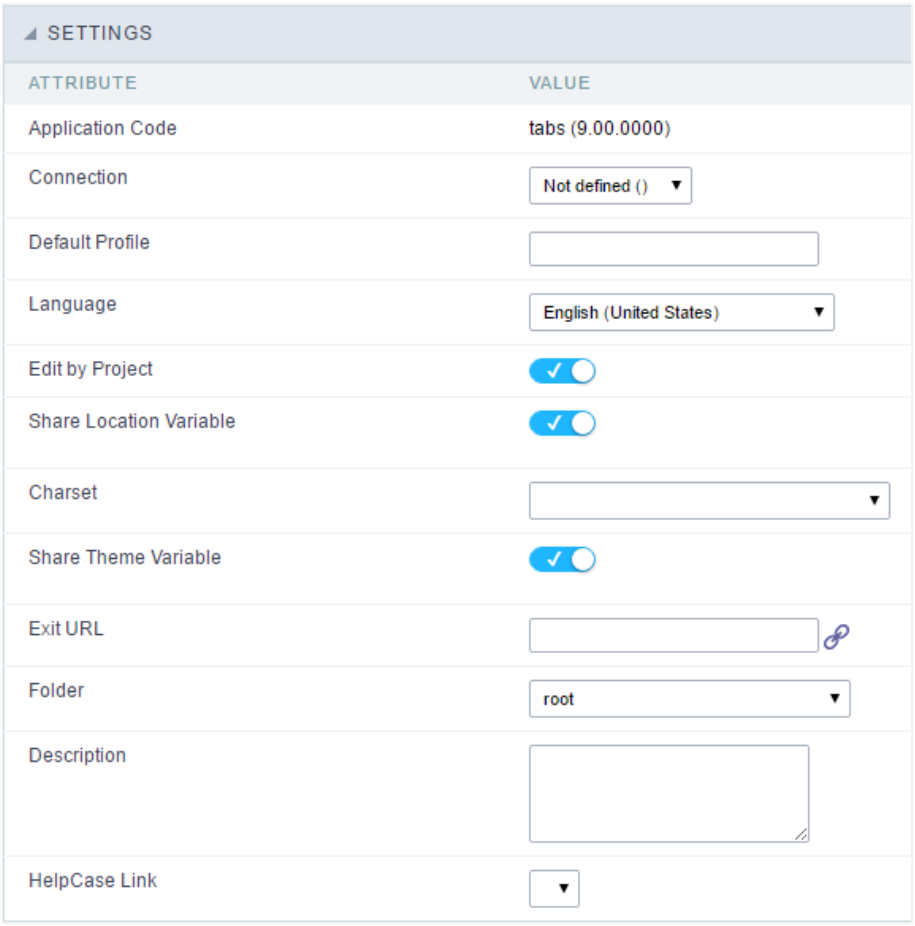
Depending on the Application, you may have more than one title option.

Related Links

Related Videos

Application Settings (Tabs Application)

General data



Settings interface.

In this folder you can set the application configuration attributes, such as:

- **Application Code** : Code for internal use of ScriptCase. Should start by a letter.
- **Connection** : Connection name of the used by ScriptCase to access the tables. The user can select it.
- **Default Profile** : Default profile to be loaded into the production environment.
- **Language** : Data formatting based on the regional settings.
- **Charset** : Share location with other applications based on the values in the session.
- **Share Theme Variable** : Shares themes with other applications based on the values in the session.
- **Exit URL** :URL to which the user will be redirected to exit the application.
- **Folder** : Folder where the application will be stored in the work group. The user can select it.
- **Description** : Application description.
- **HelpCase Link** : Using this option you can associate HelpCase files to your application.

Global variables

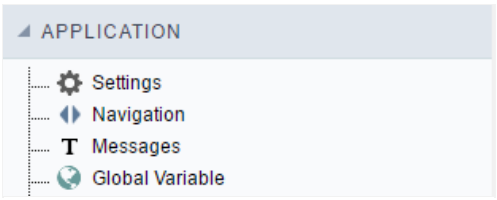
This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead

of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target="_blank"} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



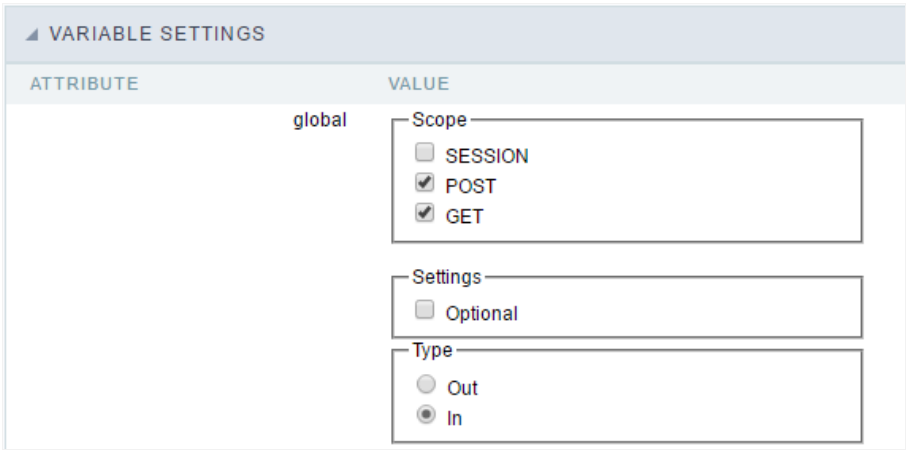
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

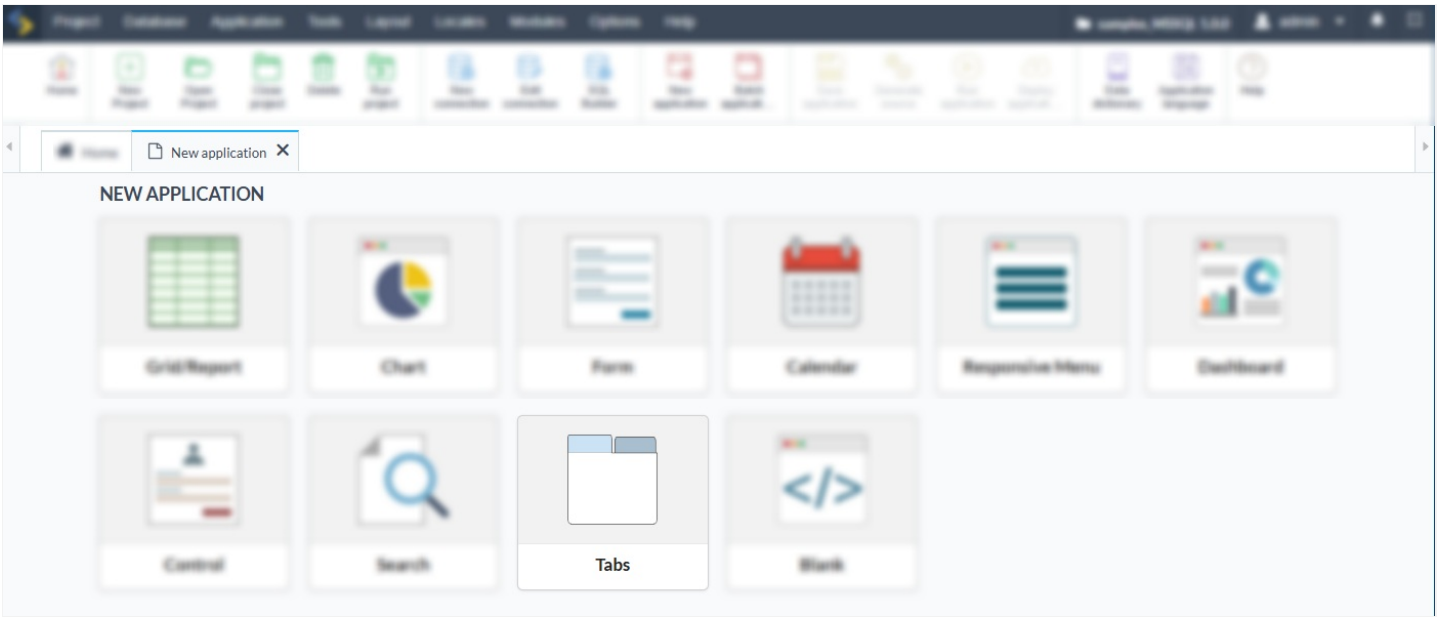
Indicates where the application is using the variable.

Creating a Tabs Application

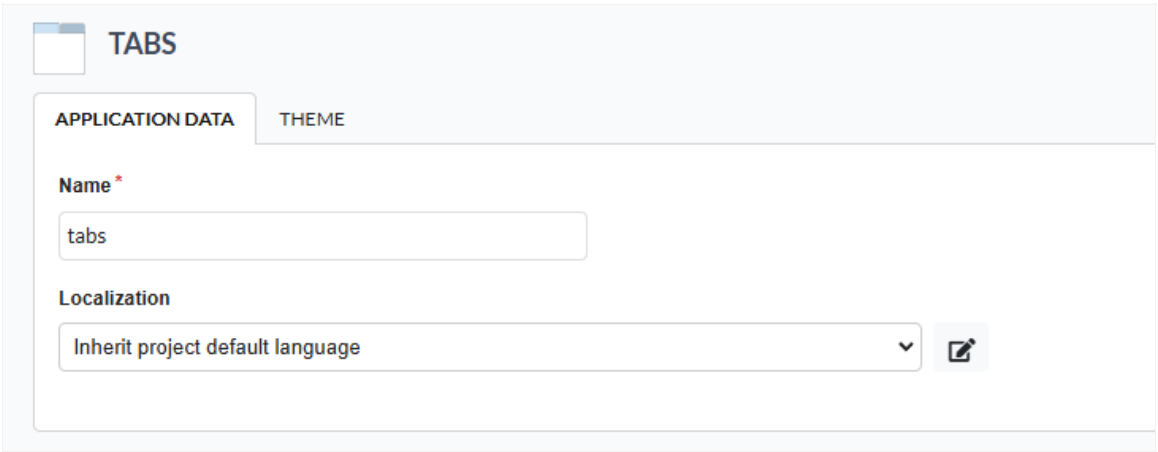
New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.



Application Data



Name

The name of the application being created. It cannot contain special characters.

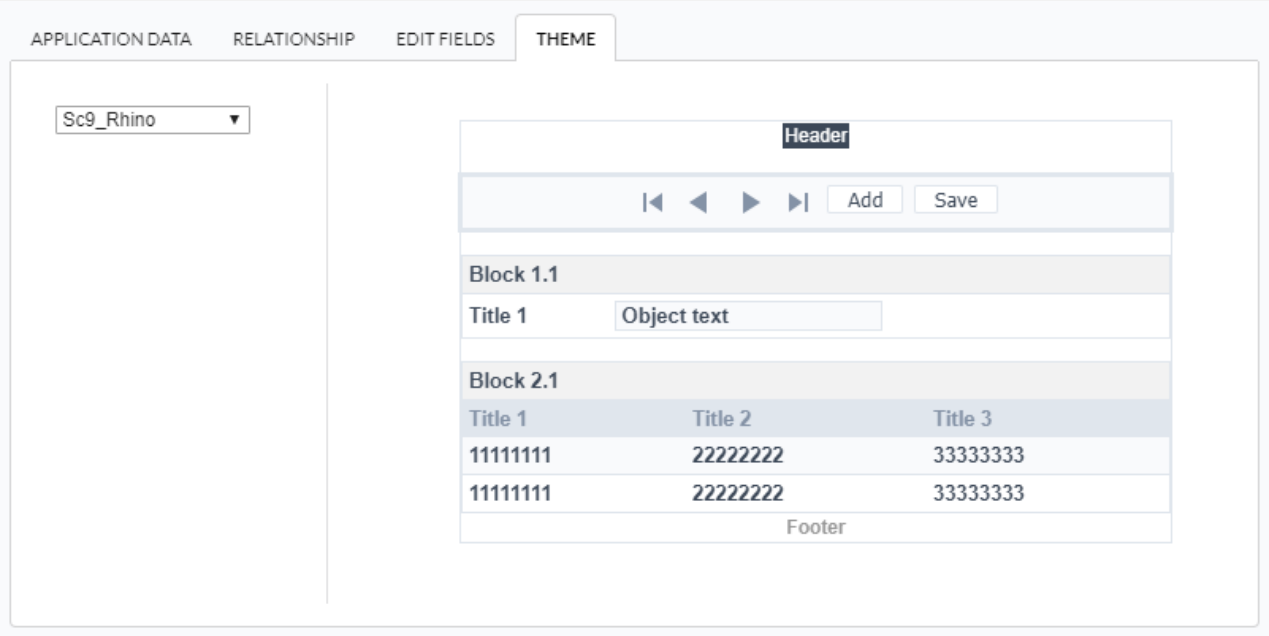
Localization

The language of the application to be created. The project's default language is automatically selected.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and

modified in **Project** > **Properties**, is automatically selected.



Related Links

Related Videos

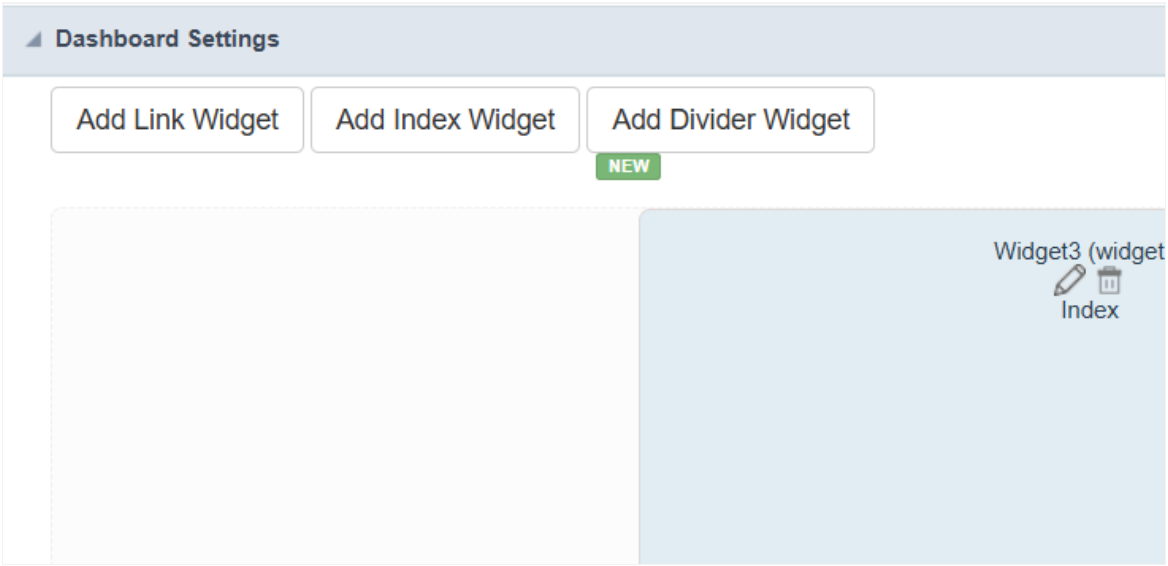
Dashboard Settings

General Overview

The Dashboard application is focused on creating components where you can visualize several applications at the same time.

With the dashboard application, it is possible to create, for example, a screen to visualize several graphs, and also, with the addition of the Index Widget create comparisons, show sum of values, among other functionalities.

On the home screen, you can also configure where the Widgets will be displayed, they can be easily dragged and mounted according to the user's need.



At the homepage you can see the Widgets and some options.

Add Link Widget

This is the option to create a link widget

Add Index Widget

This is the option for creating an index widget

Add Divider Widget

This is the option for creating an Diveder widget

And within these, there are the options for:

Pencil

Used to edit Widget settings.

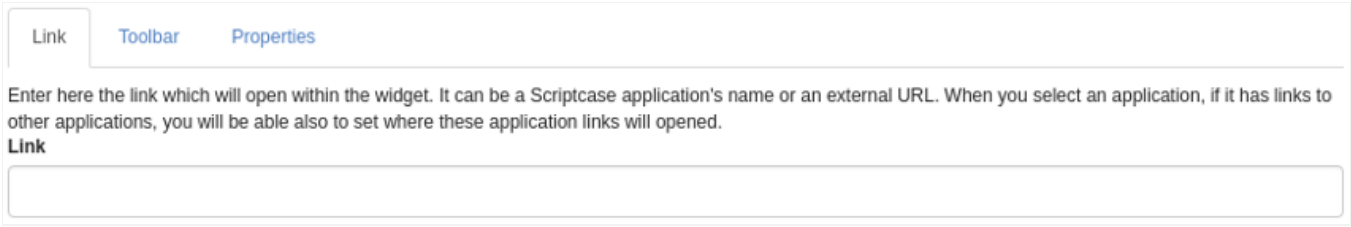
Trash

Used to Delete a Widget.

Besides that when the mouse is over the Widget, we can see a button to change the size of it.

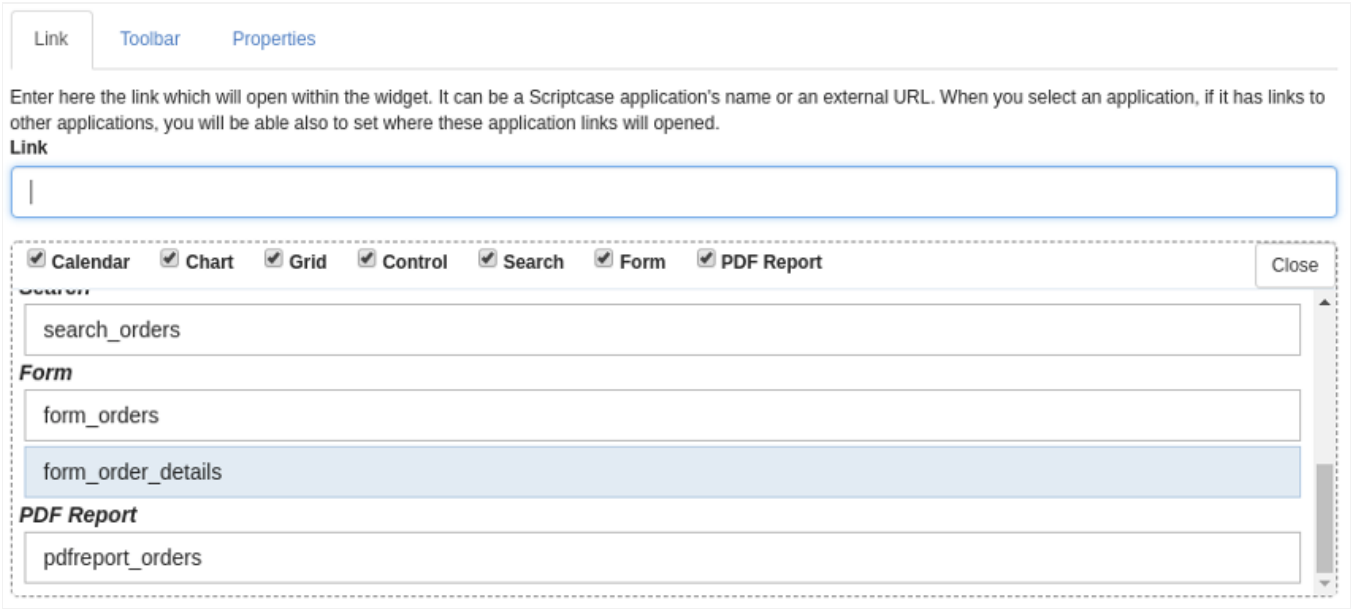
Add Link Widget

This option allows any application that exists in the project to be added to a Widget.



Link

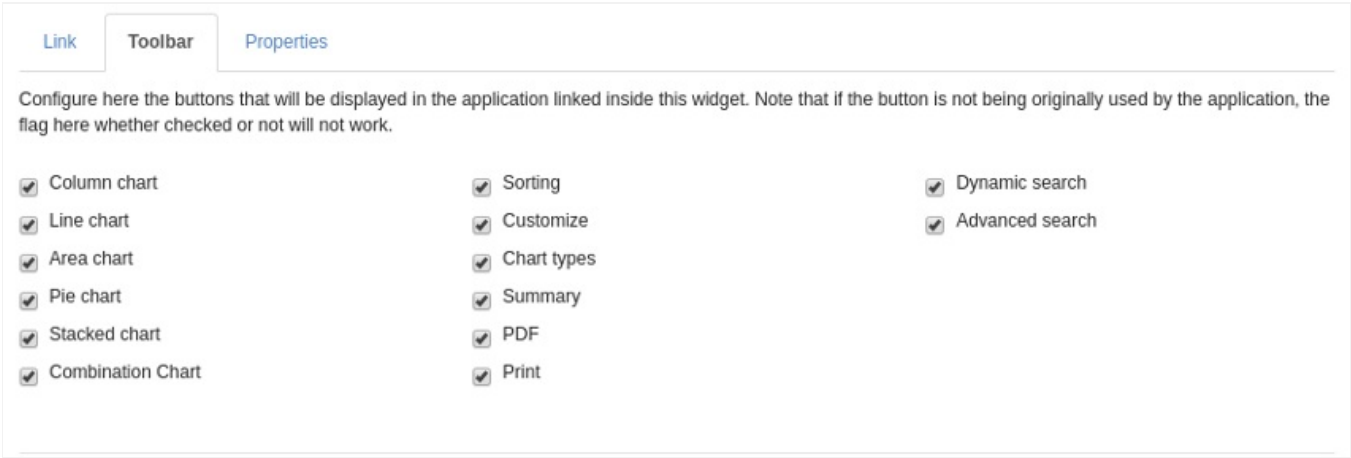
To create the link enter the Widget and the application, it is necessary only to click on the field “Connection”, then it will be shown a list with all the applications existing in the project.



Toolbar

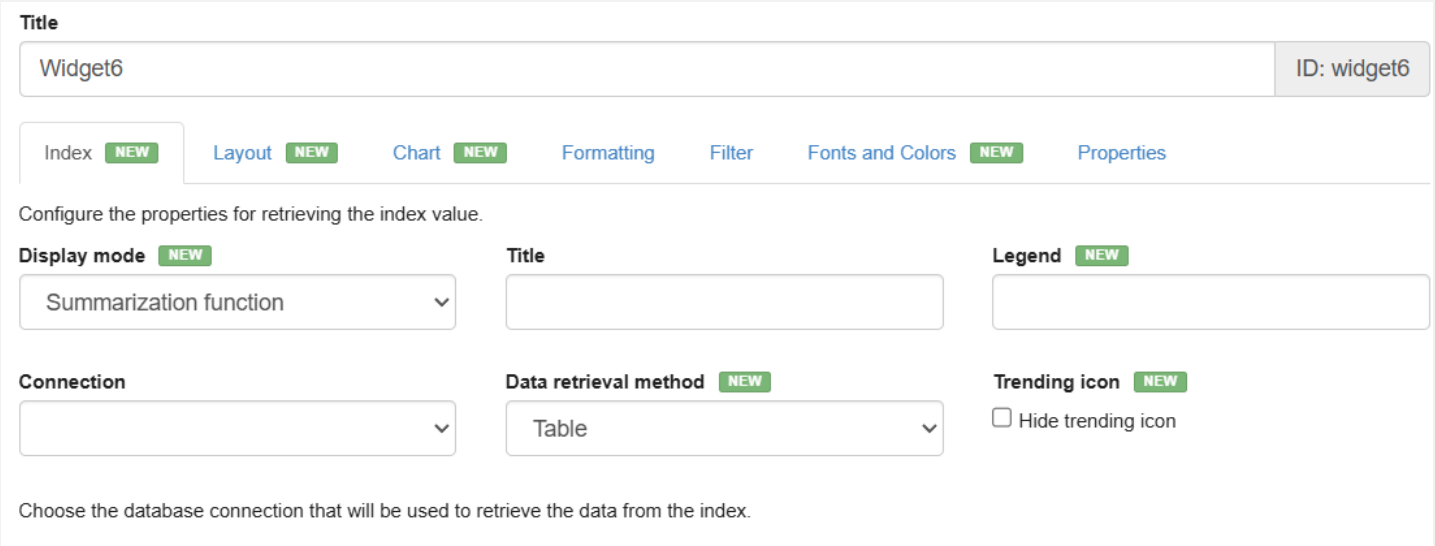
In the toolbar settings, it is possible to configure the buttons that will appear in the toolbar of the application that is configured in **connection**.

The toolbar options will be displayed according to the application being connected, that is, according to the toolbar of each application.



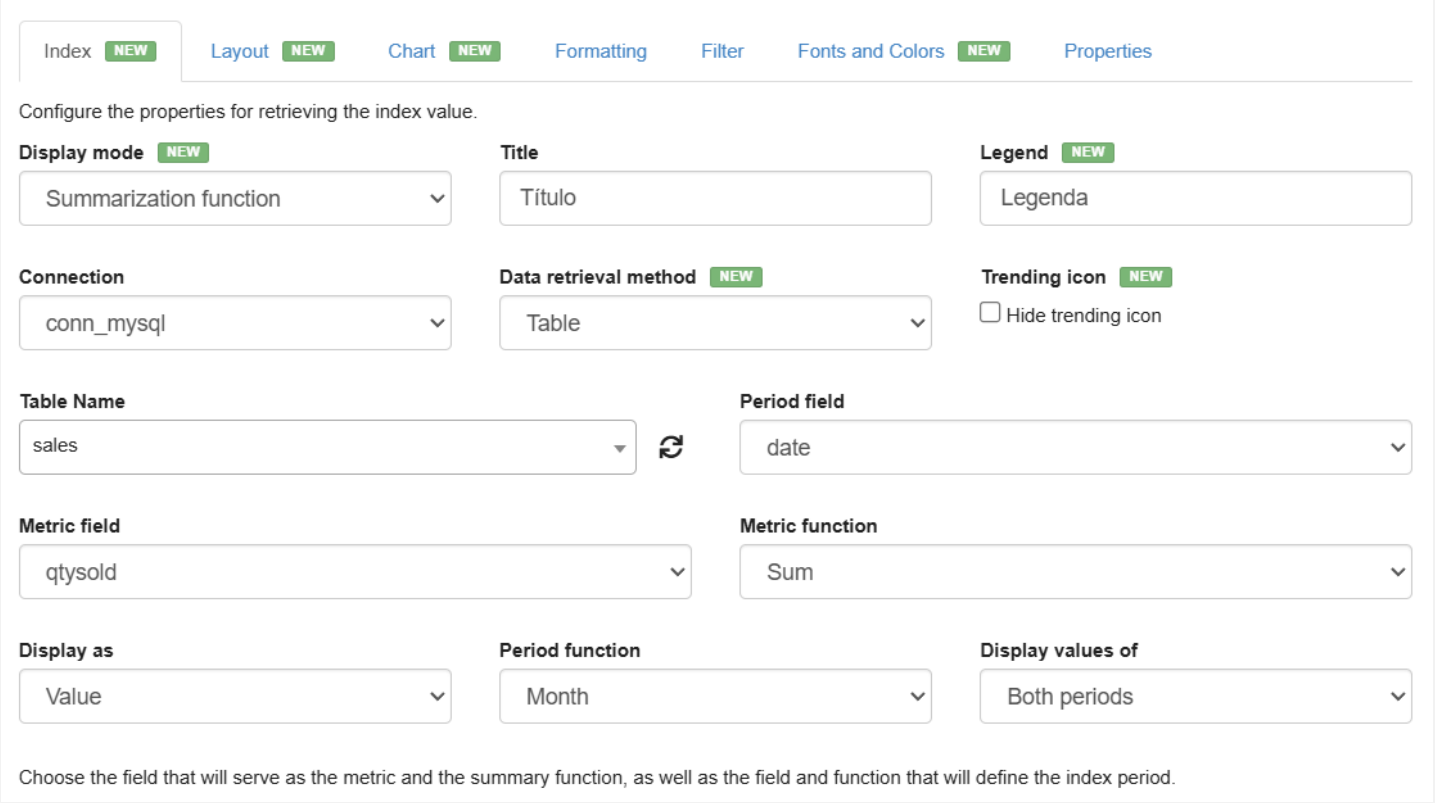
Add Index Widget

This option allows you to add to an Index Widget based on a database table.



Index

To create an Index Widget you need to make a few settings.



Display Mode

The display mode defines how the index information will be displayed. Available options include: **Summary Function and Simple Card**

Note: When the “Summary Function” display mode is selected, the Trend Icon option becomes available.

Title

This option allows you to define a title for the widget to better visualize what you want to display.

Caption

Allows you to add a caption to the information displayed in the widgets to better visualize what you want to display.

Connection

This option allows you to choose which connection will be used in the index widget; each widget can be configured with different connections.

Data Retrieval Method

Determines the method used to retrieve information from the database.

The options are:

- **Table:** allows you to select a table directly to read the data.
- **SQL Command:** can allow retrieval of custom data through the Select command.

If the select command option is selected, the Table Name, Period Field, Metric Field, and Metric Function fields will be hidden, and a field for building the select command will be displayed.

Trend Icon

When enabled, this option adds a visual trend icon (e.g., up/down arrow) next to the index value, representing growth, decline, or stability compared to a previous value.

The “Hide Trend Icon” checkbox can be selected if you do not want to display this indicator in the widget.

Table Name

This option allows you to select which table will be used in the Index Widget.

Period Field

This option allows you to select a field from the table so that the data selected in the previous option can be compared.

Metric Field

This option allows you to select a table field to create a metric, similar to the **chart** application. However, you can perform some comparisons based on the period defined in the next option.

Metric Function

Defines the metric function. The available functions are: **Sum, Average, Max, Min, Count, and Distinct Count**.

Display as

This option allows you to choose how the value will be displayed. You can choose between: Percentage Difference, Difference, and Value.

Period Function

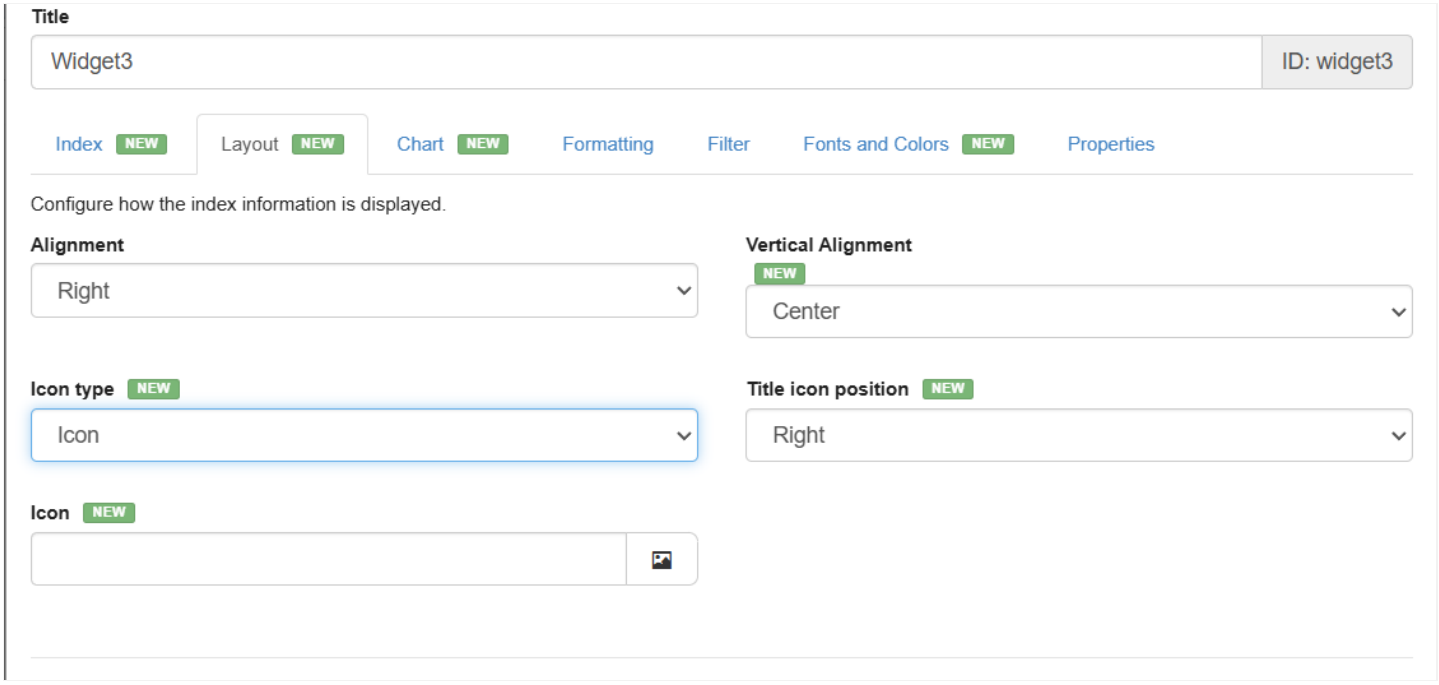
This option allows you to configure the index by defining a function for a specific period. These functions are defined by a period of time, such as: Full Date, Year, Month, Day, Half-Year, etc.

Display values of

This option allows you to display values for:

- **Both periods**
- **Current period**
- **Last period**
- **None**

Layout



Alignment

Allows you to set the horizontal alignment of the content within the widget. Options typically include:

- **Left:** aligns the content to the left margin.
- **Center:** centers the content horizontally.
- **Right:** aligns the content to the right margin.

Vertical Alignment

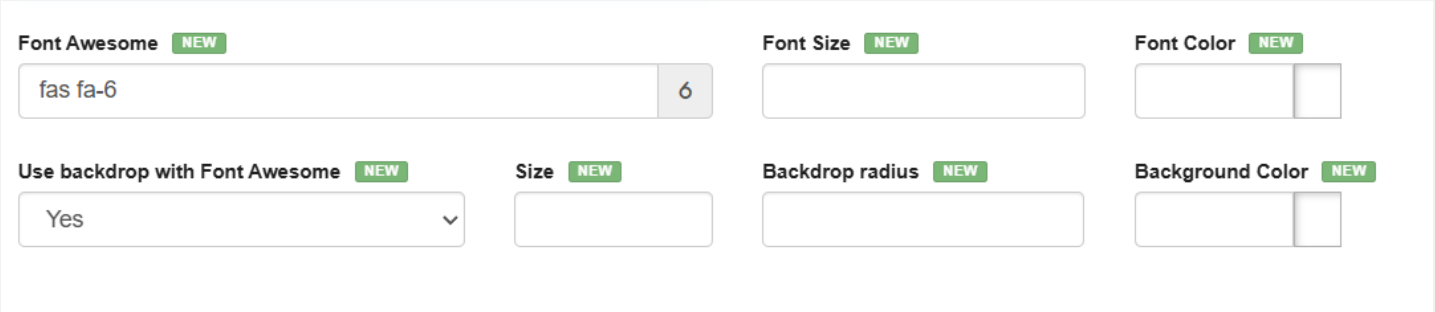
Controls the vertical positioning of the information within the widget. Can be set to:

- **Top:** positions the content at the top of the widget.
- **Center:** positions the content in the vertical center.
- **Below:** positions the content at the bottom.

Icon Type

Allows you to choose the type of icons that will be used in the widget. Options include:

- **Icon**: With this option selected, the icon field will be displayed so you can choose an icon directly from the image manager.
- **FontAwesome**: With this option set, you will see a series of fields available for configuring the FontAwesome icon.



Title Icon Position

Defines where the title icon will be displayed in relation to the text. Available options include:

- Above
- Left of the title
- Right of the title
- Left
- Right

Chart

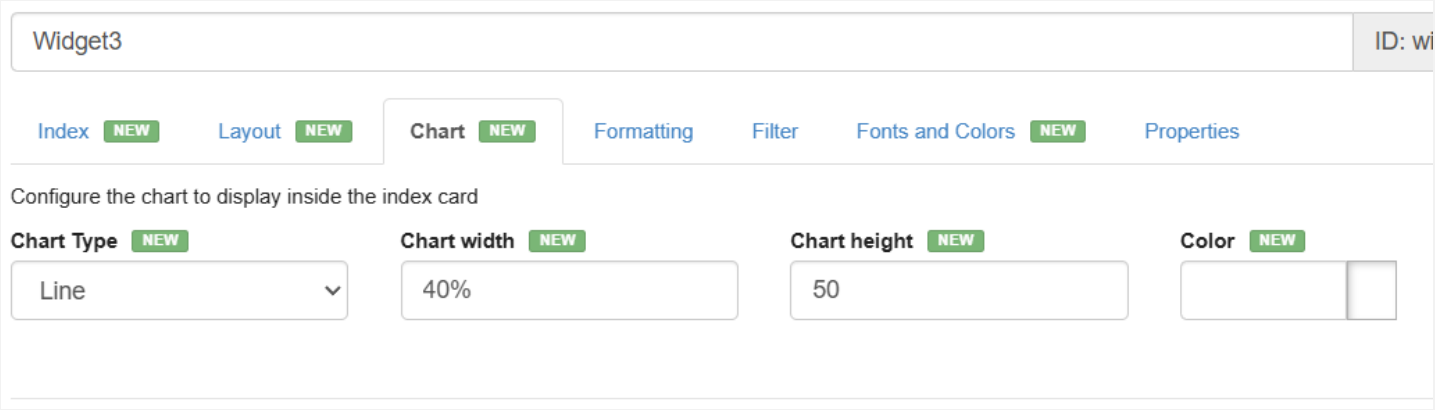


Chart Type

This option enables chart display in indexes. You can choose from three chart types: Line, Area, and Column.

If you don’t want to display the chart, simply select “None.”

Here’s an example of a line chart added to the widget:

Chart Width

Allows you to set a percentage value for the chart width.

Chart Height

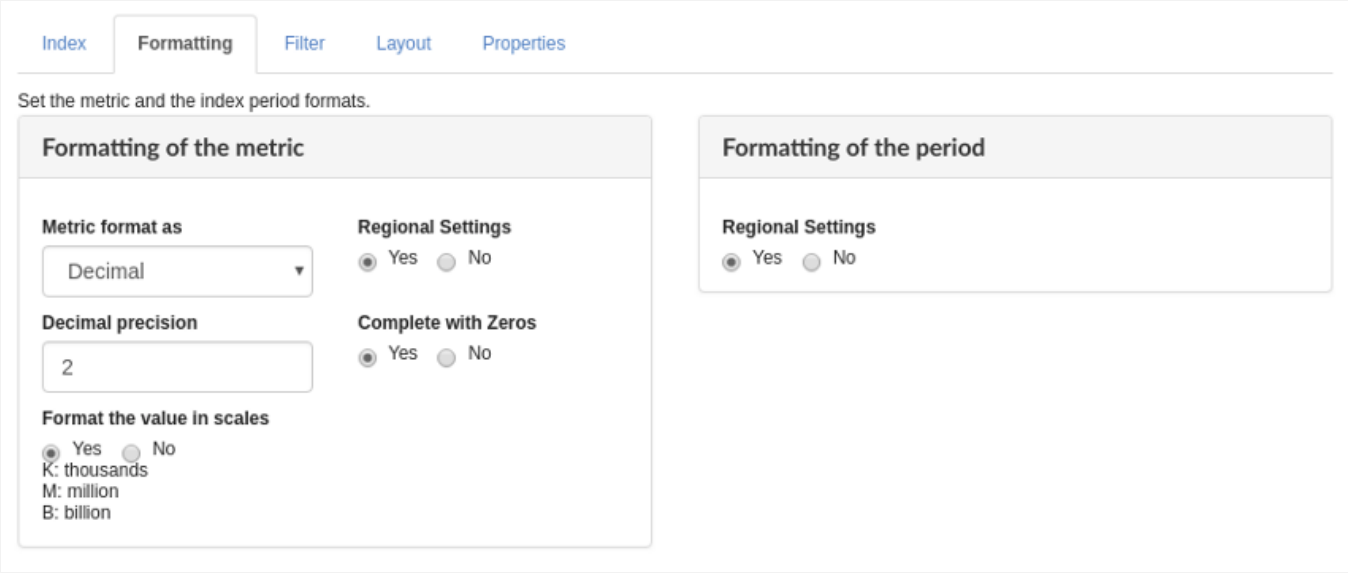
Allows you to set a value for the chart height.

Chart Color

Allows you to choose the chart color.

Formatting

In this option you can configure the formatting of the metric and period that will be shown in the execution of the filter.



Metric Formatting

Metric Formatting as

Allows you to define the type of data that the metric will be shown, there are three options:

Internal: The index metric will be shown as a value of integer type. **Decimal**: The index metric will be shown as a value of type Decimal. **Currency**: The metric of the index will be shown as a value of type Modal.

Use regional settings

Lets you define whether the metric will follow the settings that were defined in the Regional Settings.

Decimals

Allows you to set how many decimal places will be in the metric index, however this will only be available when the metric format is **Decimal** or **Currency**.

Complete with zeros

Allows you to set whether the value will be completed with zeros when set to **Decimal** or **Currency**.

Format the value in scales

Lets you set whether the number in the index will appear in a scale or the whole number. If **Yes** is chosen, the number instead of **10.000** will appear with **10 k** value.

Period Formatting

The options are shown according to the setting in **Feed Function** which is set to **Index**, so the setting can vary between:

Use regional settings

Allows you to set whether the period will follow the settings that were set in the regional settings.

Display the name of the month

Lets you to set whether to display the name of the month.

Prefix for semester, four-month, quarter, two-month, week.

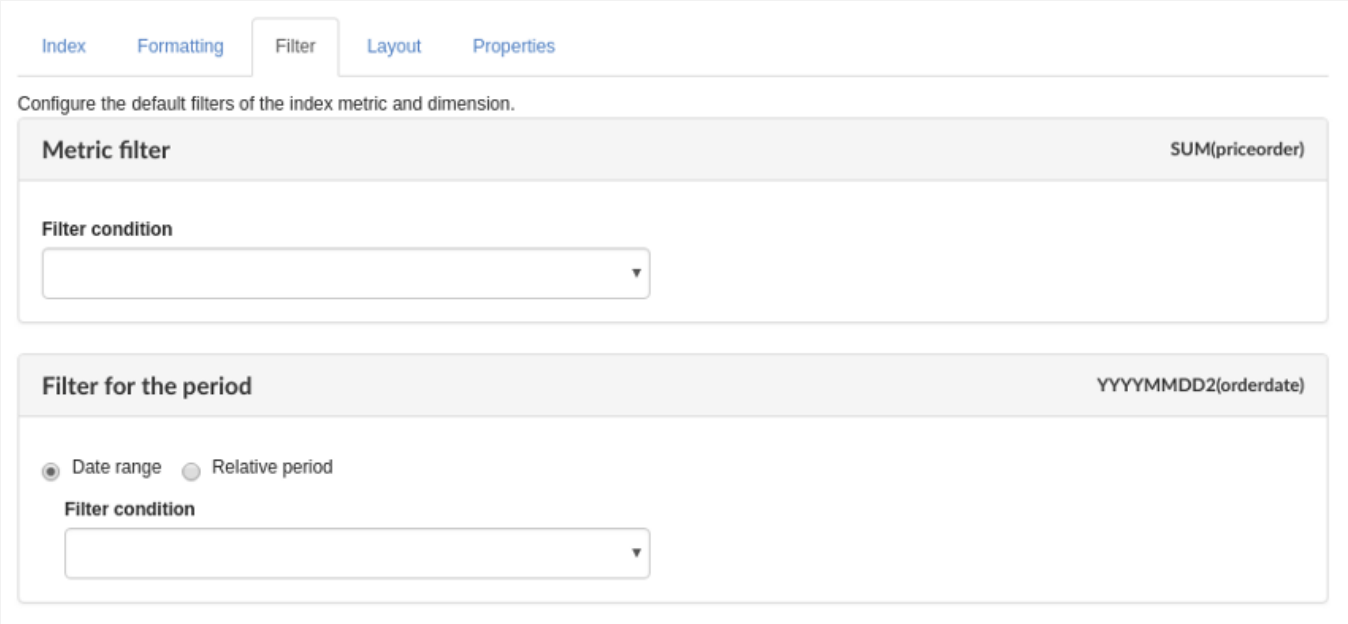
Lets you inform a lang or a title that the developer wants to appear.

Display name of the day

Lets you set whether to show the name of the day.

Filter

Allows you to perform the filter configuration that will be displayed when the Index Widget is running.



Metric filter

Filter conditions

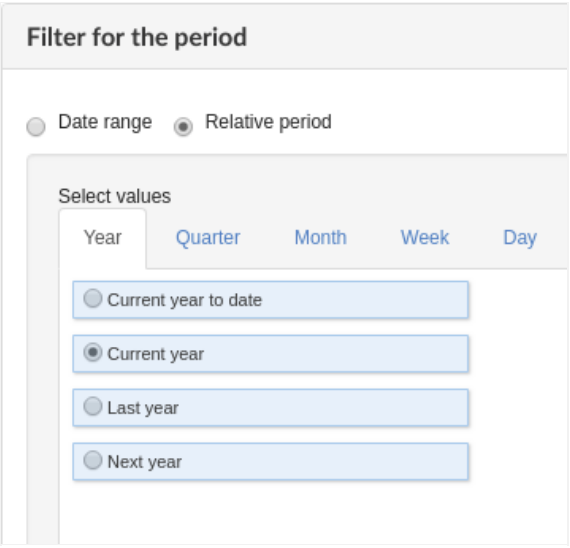
In this option we can define the condition of the filter so that the information regarding that metric is shown, and it is not possible to change that at run time.

Period Filter

Date Ranges

In this option we can define the condition of the filter so that the information regarding that period is shown, and it is not possible to change at run time.

Relative period



We must define which periods are available for use in the filter.

Another option referring only to the relative period is the possibility of creating conditions with relative periods customized

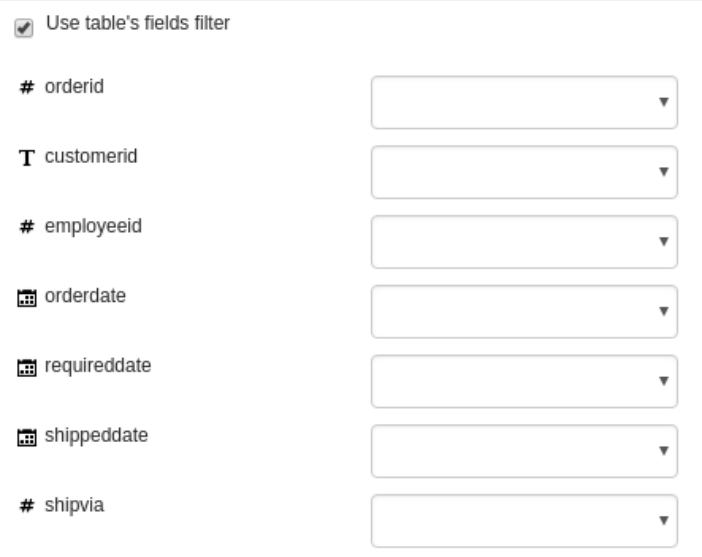
directly within the filter interface.

The available values are separated by tabs:

Year: Sets the year used in the filter. **Quarter**: Defines the quarters used in the filter. **Month**: Sets the usage of the months of the year. **Week**: Sets the use of the weeks of the year. **Day**: Sets the use of the days of the month.

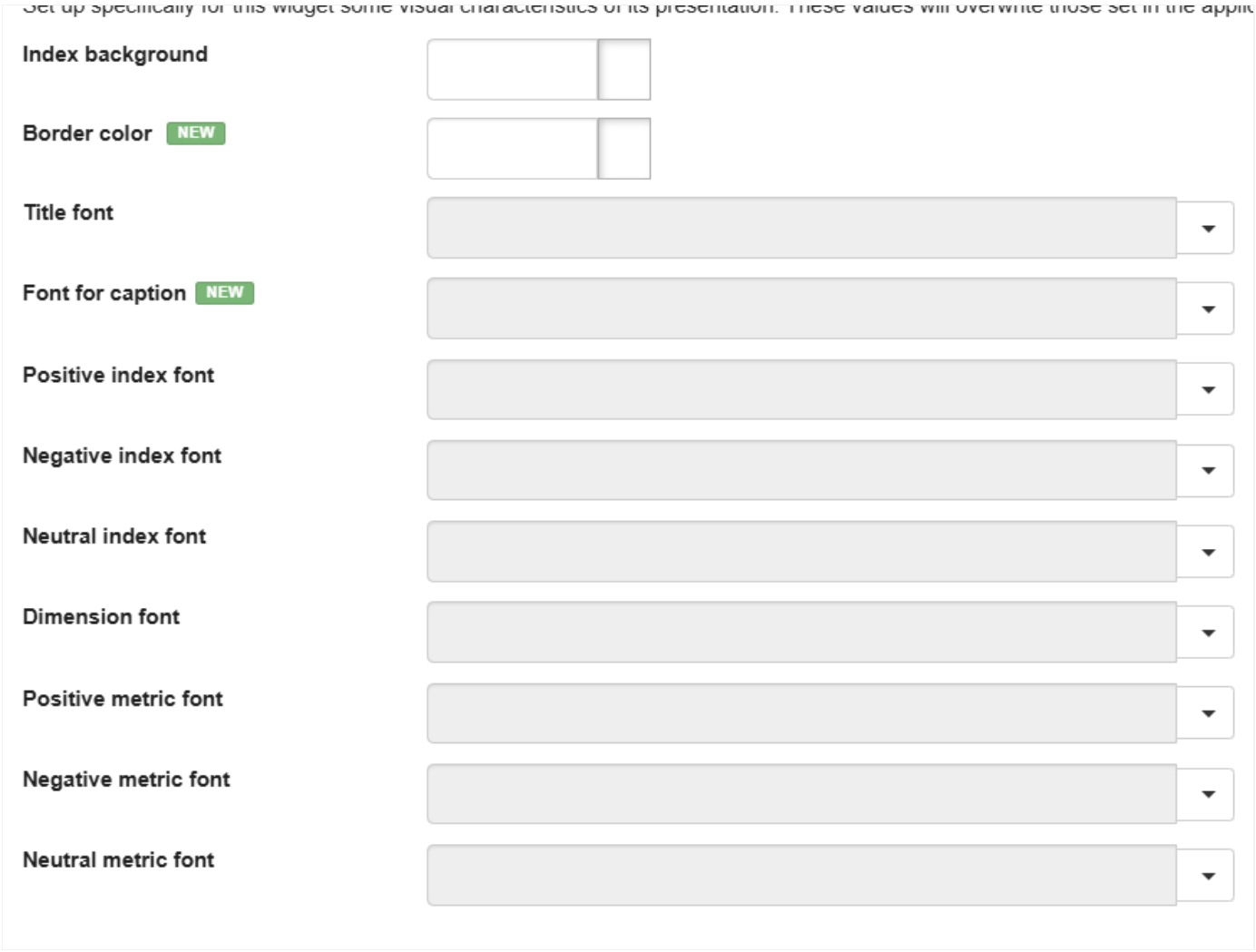
Use field table filter

Allows you to create filters from table fields.



Field and Colors

In this option you can configure the layout of the map that will appear in the Widget.



Index background

It allows you to set a background color for the widget content.

Boder Color

It allows you to set a color for the widget border

Title font

Allows you to configure the font to be displayed in the index title, in this option there are four more options:

Family/source type Font size Color of the font The font style

Font for caption

It allows to set the font for the widget caption. In this option there are four more options:

Family/source type Font size Color of the font The font style

font of positive index

Allows you to configure the font that will be displayed in the index when positive, in this option there are four more options:

Family/source type Font size Color of the font The font style

Negative index source

Allows you to configure the font that will be displayed in the index when negative, in this option there are four more options:

Family/source type Font size Color of the font The font style

Source of the neutral index

Allows you to configure the font to be displayed in the index when neutral, in this option there are four more options:

Family/source type Font size Color of the font The font style

Source of the dimension index

Allows you to configure the font size, in this option there are four more options:

Family/source type Font size Color of the font The font style

Source of positive metrics

Allows you to configure the font that will be displayed in the metric when positive, in this option there are four more options:

Family/source type Font size Color of the font The font style

Negative metric source

Allows you to configure the font that will be displayed in the metric when negative, in this option there are four more options:

Family/source type Font size Color of the font The font style

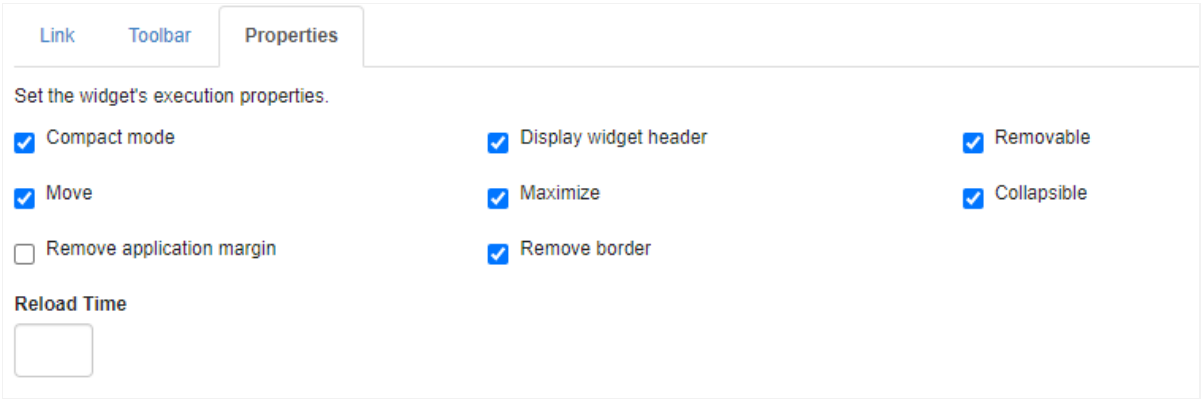
Neutric metric source

Allows you to configure the font that will be displayed in the metric when neutral, in this option there are four more options:

Family/source type Font size Color of the font The font style

Properties

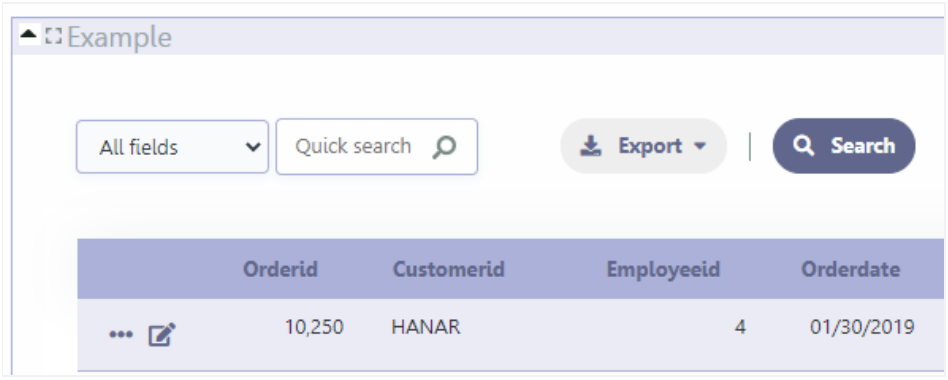
Allows you to configure some features that will be available at execution time, they are



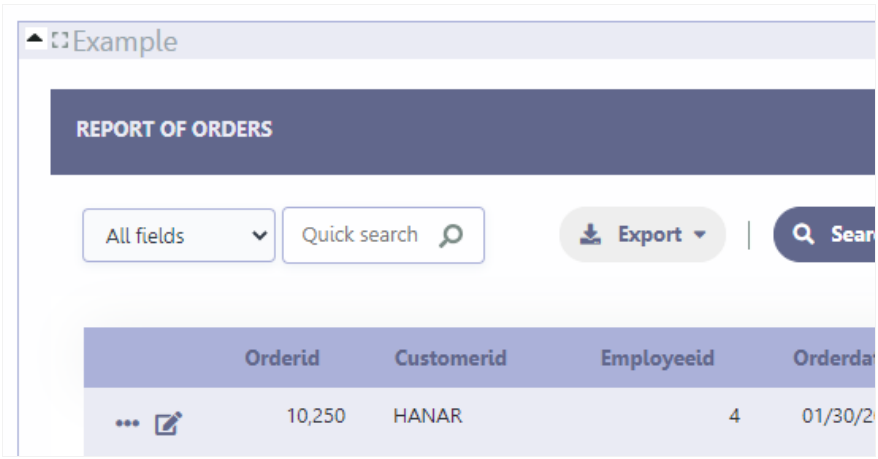
Compact Mode

Removes header and footer of applications added within the Widget:

Examplewith compact mode enabled

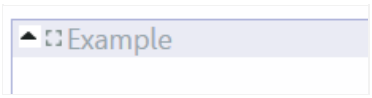


Example with compact mode disabled



Show widget header

Allows you to enable or disable the running widget header. The title and **Remove**, **Mover**, **Maximizar** and **Expand**__ options will be displayed, according to the individual configuration of the options.



The **Remove**, **Mover**, **Maximizar** and **Expand** options are only available if the header is being displayed.

Remove

Enables or disables the button to delete the widget at run time.

The widget will be displayed again when reloading the page.

Mover

Enables or disables the option that allows widgets to be moved at run time.

Maximize

Enables or disables the option to maximize the widget.

Expand

Enables or disables the button to expand the widget.

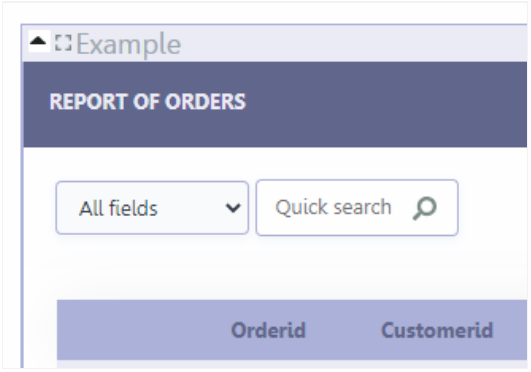
Reload time

Allows you to define a time in seconds for the Widget to be reloaded, thus reloading the information.

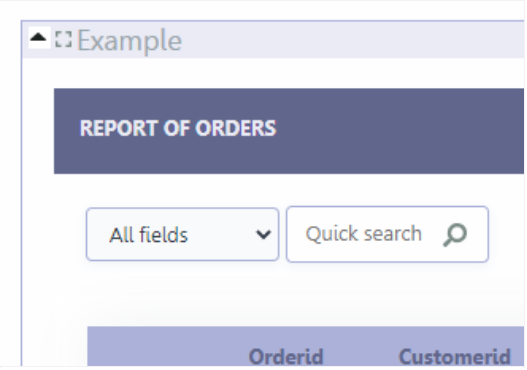
Remove application margin

This option removes application margins within the Widget.

Example of the application with the option remove margin enabled



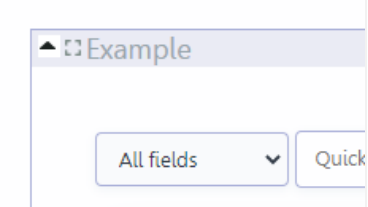
Example of the application with the option remove margin disabled



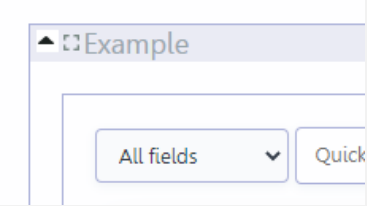
Remove border

This option removes the edges of the applications used within the Widget

Example of the application with the remove border option activated



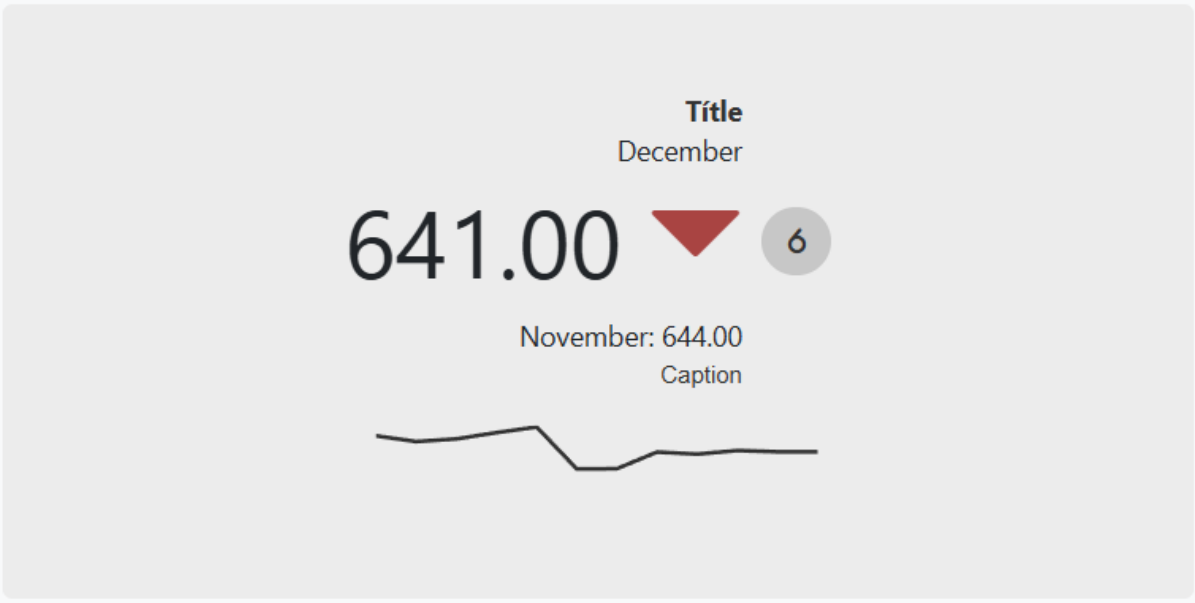
Example of the application with the remove border option disabled



Divider Widget

The divider widget serves as a divider between widgets. This creates a divider between widgets, improving visualization.

Example of using the divider widget to separate widgets:



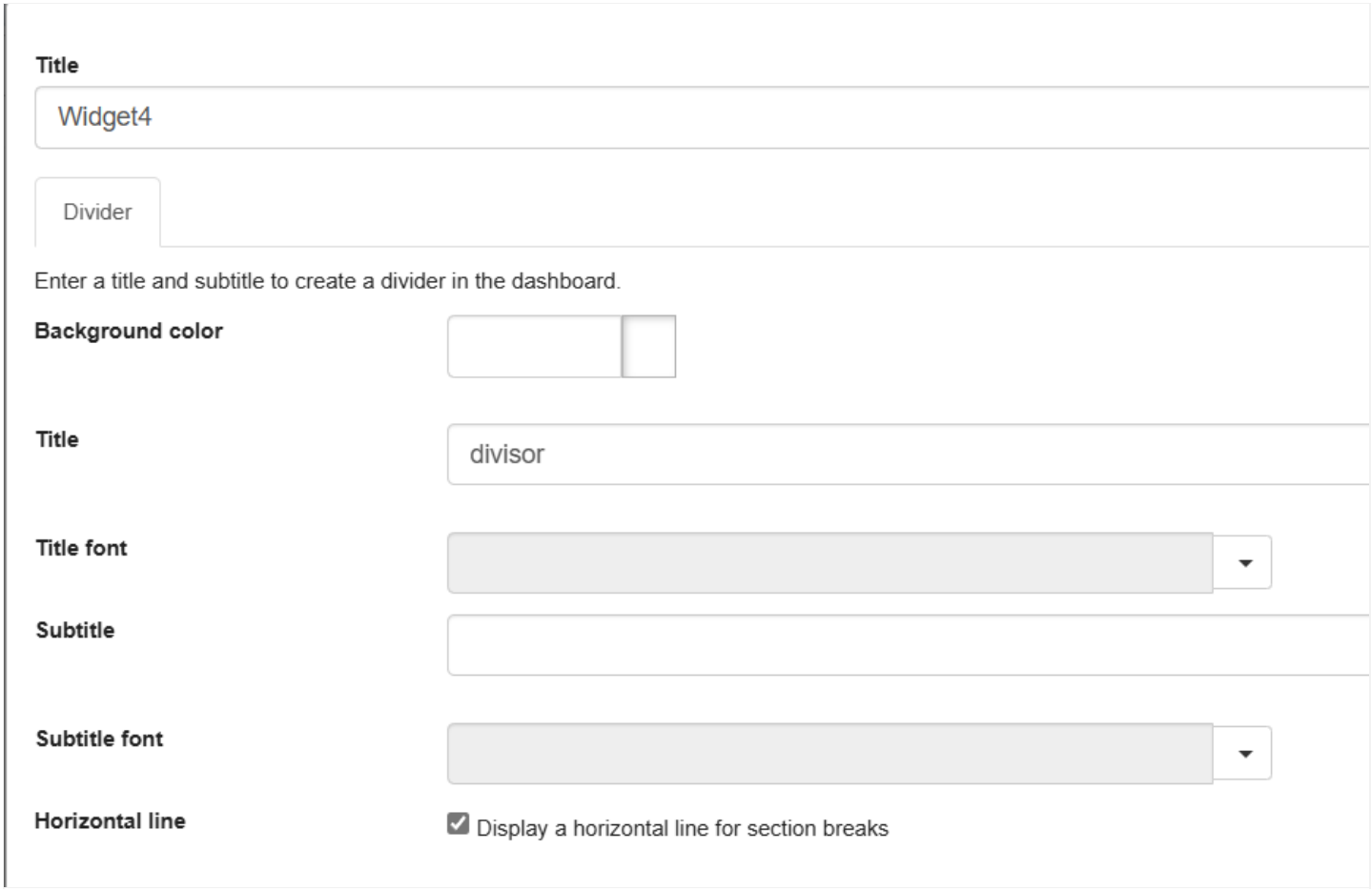
divisor

Widget5

Quick search | Columns | Sorting | Group By | Expo

	Customerid	Companyname	Contactname	Con
...	ALFKI	Alfreds Futterkistes1	Alice	Sale
...	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owr
...	ANTON	Antonio Moreno Taqueriila	Antonio Moreno	Owr
...	AROUT	Around the Horn	Thomas Hardy	Sale

Divider



Background Color

Sets the background color of the divider area. This color is applied behind the title and subtitle, helping to visually highlight the section on the dashboard.

It can be used, for example, to create colored bands that separate thematic areas of the dashboard.

Title

Field used to define the main title of the divider.

Title Font

Allows you to select the typographic style applied to the title. You can adjust the font type, size, weight (bold), and other characteristics according to the options available in the project.

The available options are:

Font family/type Font size Font color Font style

Subtitle

Field for additional text that appears immediately below the title. The subtitle can contain additional descriptions, notes, or contextual information about the section's content.

Example: Data from the last quarter or Updated daily at 8:00 AM

Subtitle font

Sets the typographic style of the subtitle, allowing you to adjust the text's appearance according to your desired design.

The available options are:

Font family/type Font size Font color Font style

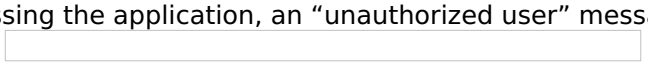
Horizontal line

When you select this option, the system inserts a horizontal line just below the title.

Related Links

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application. 

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Headers

Desabilitar o Auditor XSS

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

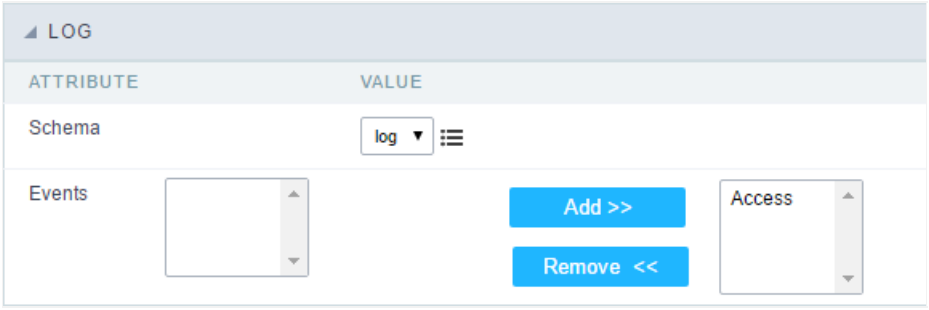
Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Dashboard Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

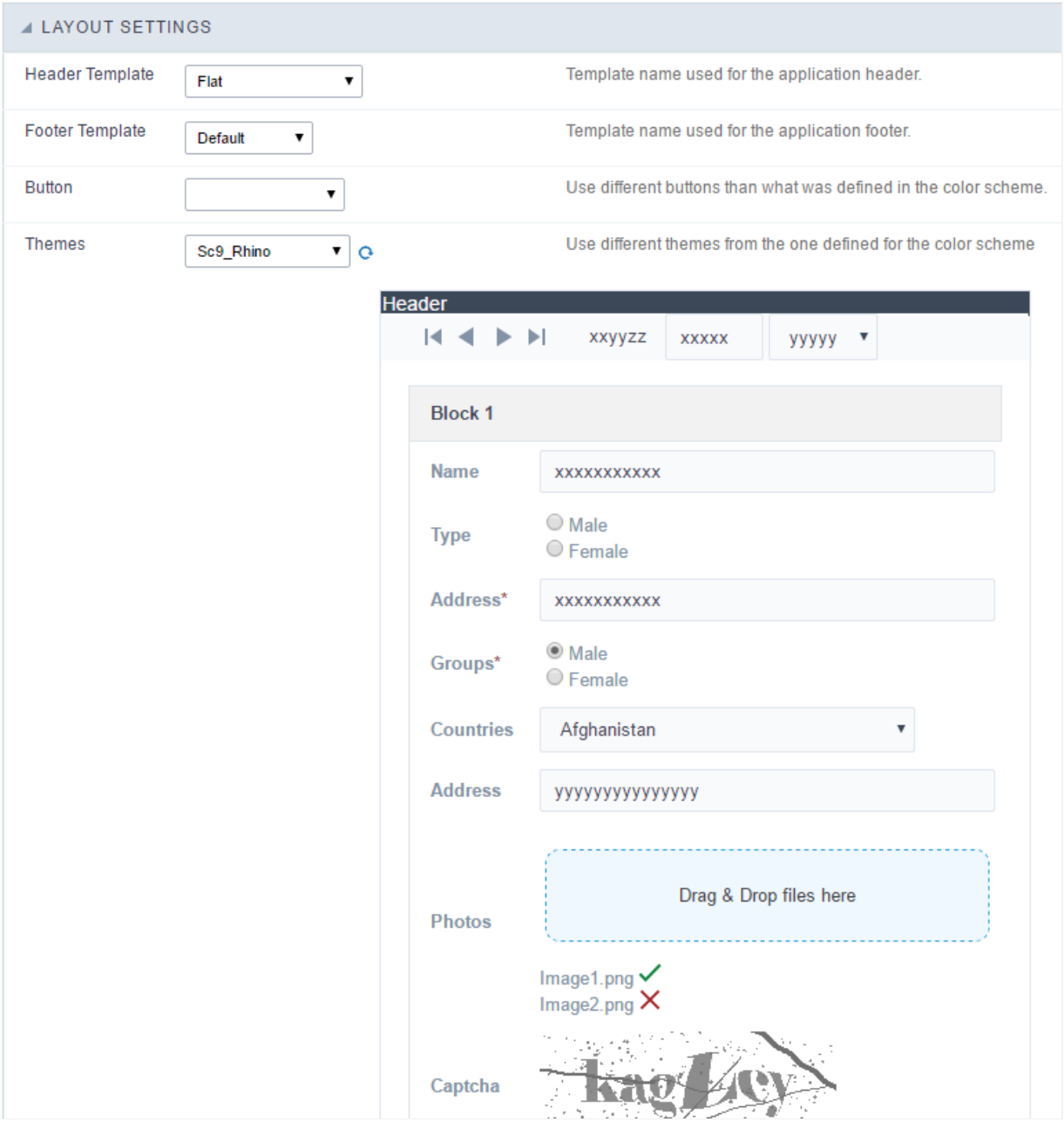
Events You can select just specific events to save in the log.

Dashboard Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout Settings

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

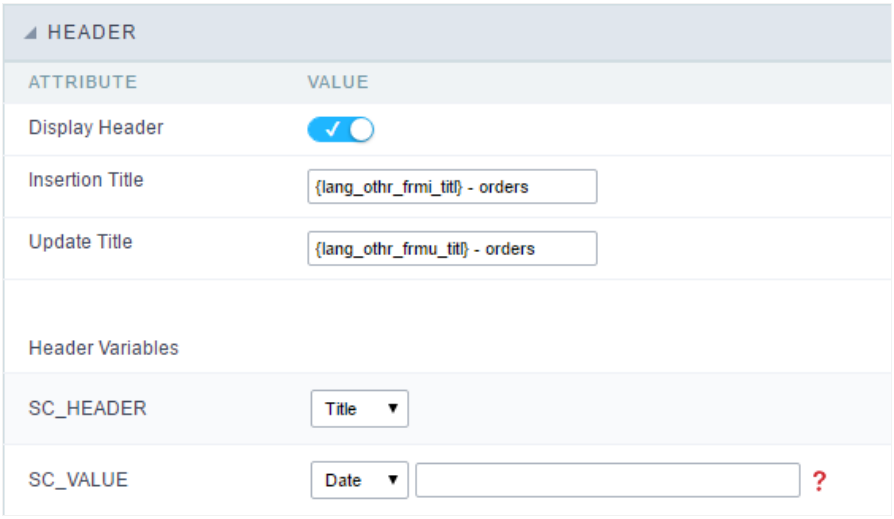
Allows choosing the button theme for the application.

Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

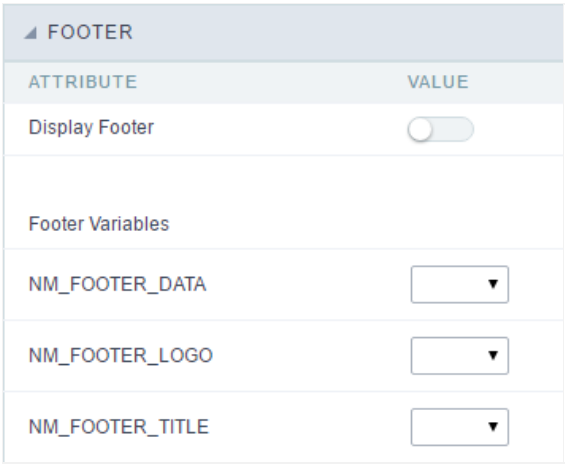
Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

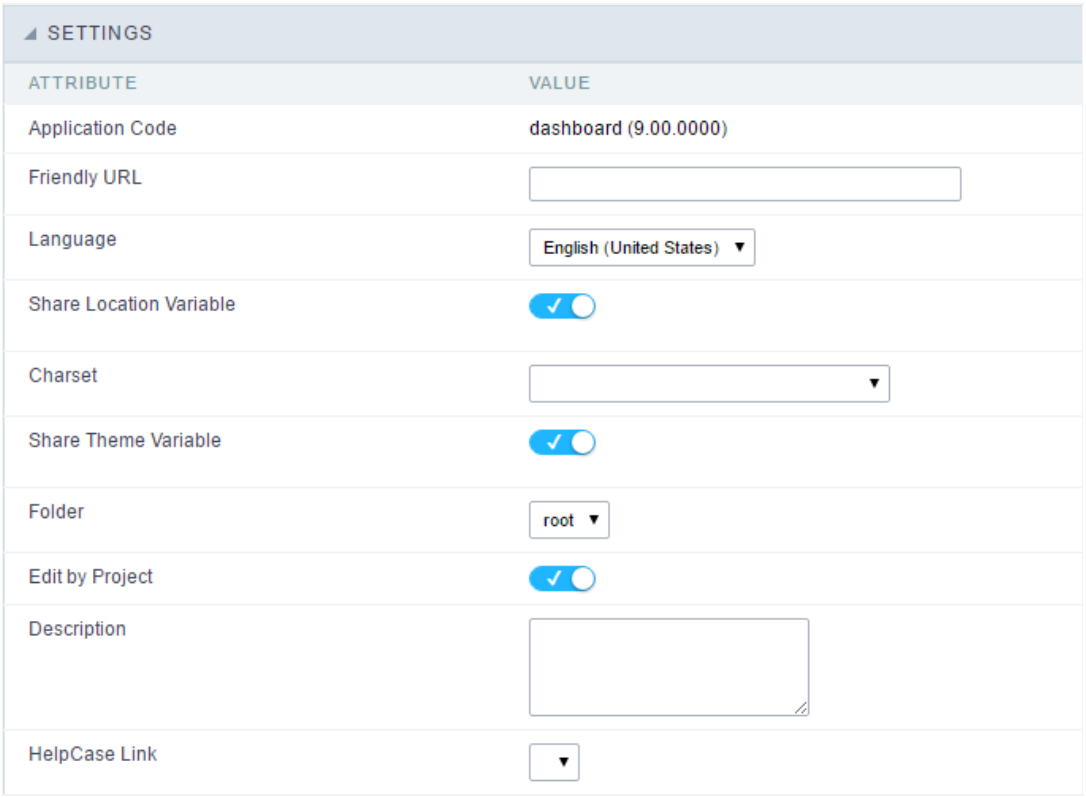
Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Application Settings (Dashboard)

General data



Settings interface.

In this folder you can set the application configuration attributes, such as:

- **Application Code** : Code for internal use of ScriptCase. Should start by a letter.
- **Connection** : Connection name of the used by ScriptCase to access the tables. The user can select it.
- **Default Profile** : Default profile to be loaded into the production environment.
- **Language** : Data formatting based on the regional settings.
- **Charset** : Share location with other applications based on the values in the session.
- **Share Theme Variable** : Shares themes with other applications based on the values in the session.
- **Exit URL** :URL to which the user will be redirected to exit the application.
- **Folder** : Folder where the application will be stored in the work group. The user can select it.
- **Description** : Application description.
- **HelpCase Link** : Using this option you can associate HelpCase files to your application.

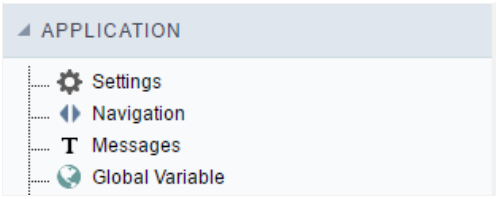
Global variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target="_blank"} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



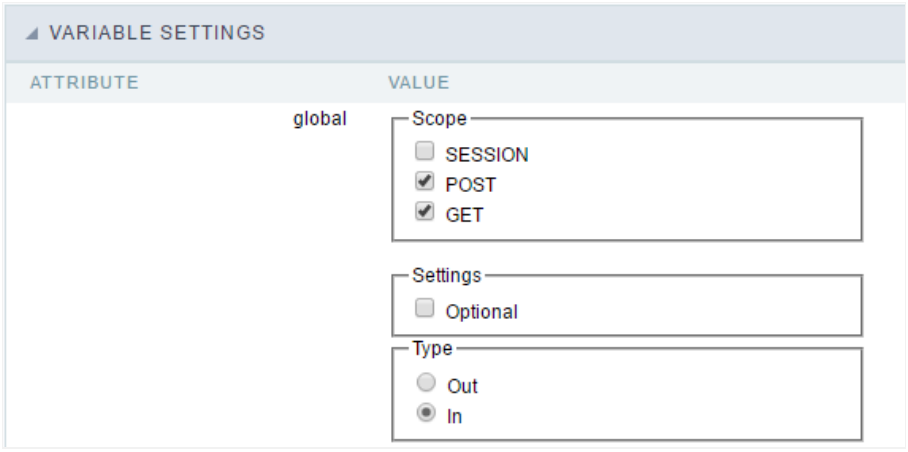
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

Indicates where the application is using the variable.

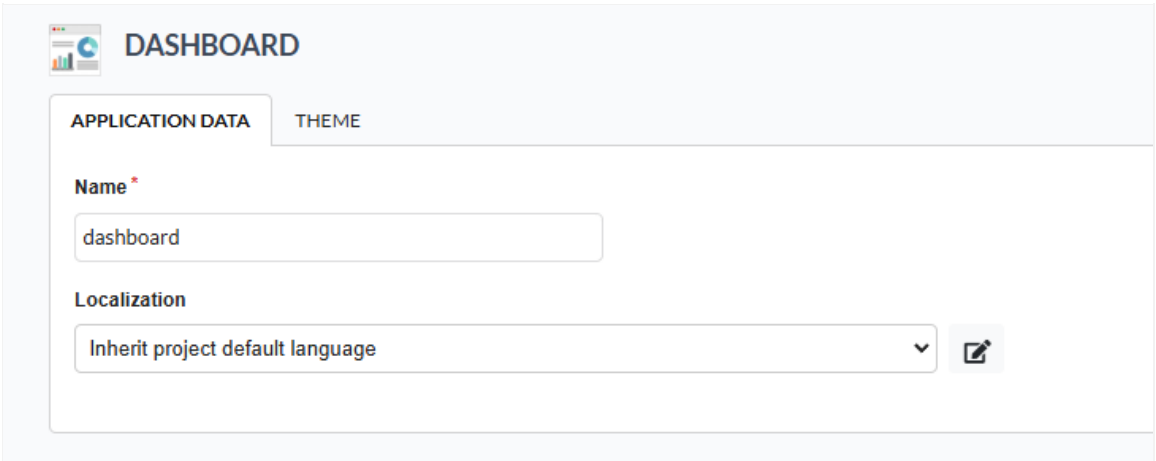
Creating a Dashboard Application

New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.

Application Data



Name

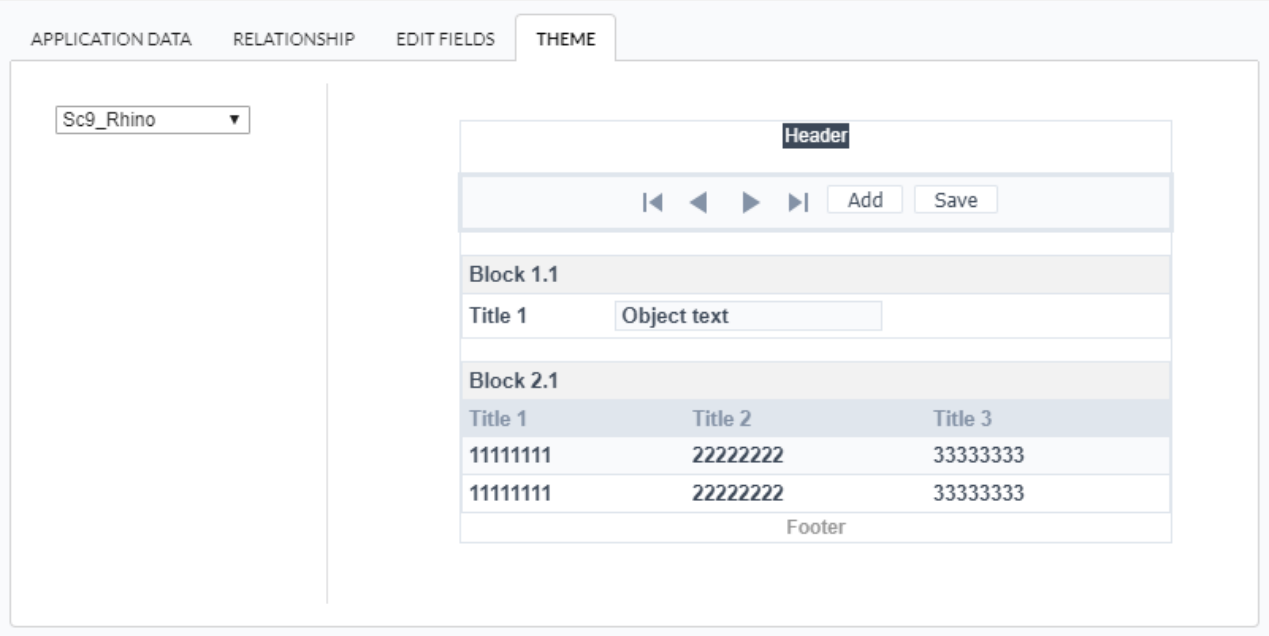
The name of the application being created. It cannot contain special characters.

Localization

The language of the application to be created. The project's default language is automatically selected.

Theme

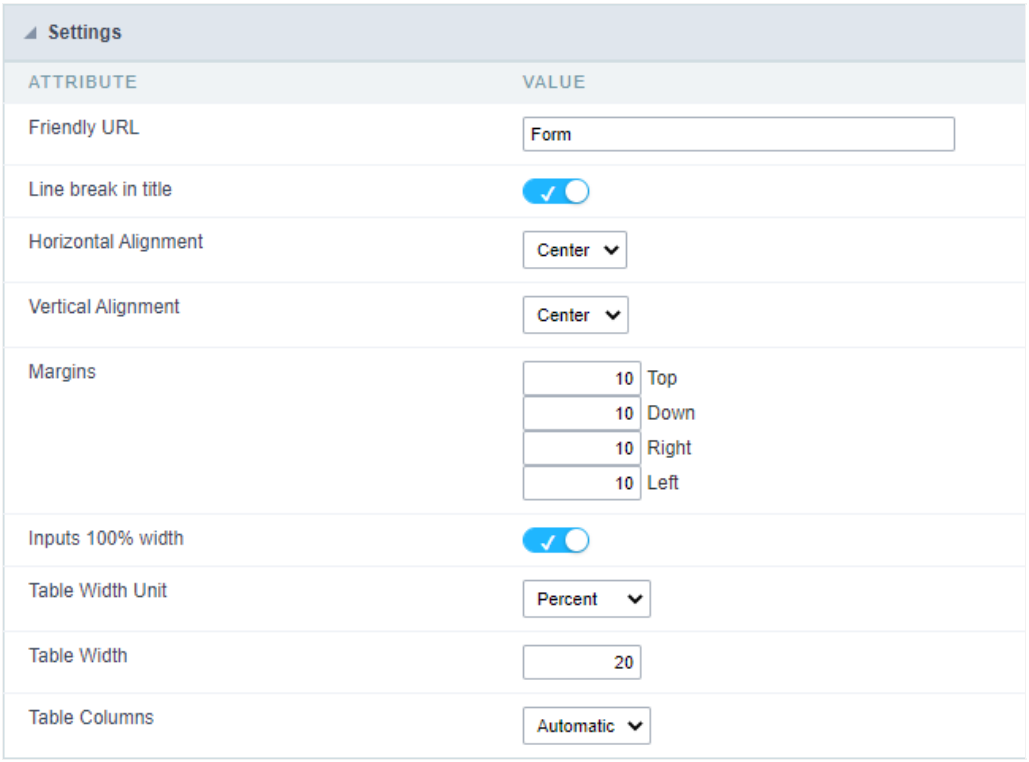
On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.



Related Links

Control Settings

This interface is useful for editing the main display settings of the app.



Form Settings Interface.

Friendly URL

Allows to define a URL for the application different from the application name. You can use the following chars (a-z, A-Z, 0-9, -_). You still can change it on the initial screen of the home project, through the “Friendly URL” column in the apps list.

Line break in title

Use it if you want to break the line on the field titles.

Horizontal Alignment

Allows you to set the alignment of the application on the page.

Vertical Alignment

Allows you to define the initial vertical alignment of the application (Above, Centered and Below).

Margins

Allows to define the margins of the application (Right, Left, Up and Down) in pixels.

Table Width

The width of the form table. Scriptcase uses Plain HTML to generate applications by using tables lines and cells.

Table Width Unit

Measurement unit for the table width defined in the previous option, being: percentage, pixel, or automatic.

Table Columns

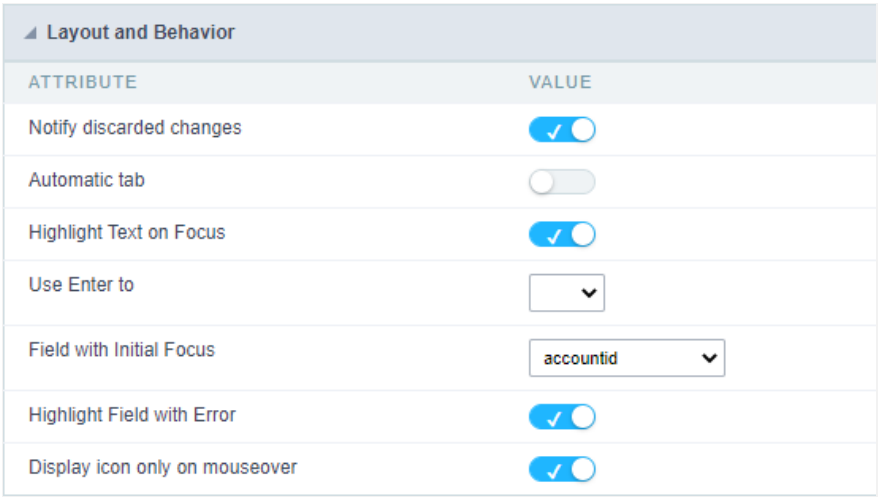
This parameter defines the column (fields) width of the table (application).

Labels width

When the previous option is set to “Provided” you must inform the width of the labels here.

Layout and Behavior

This interface allows setting the behavior of the app.



Layout and Behavior configuration Interface.

Notify discarded changes

Notifies the user when any changes made will be lost when reloading the data.

Automatic tab

Changes the focus to the next field when the amount of characters reaches its defined limit.

Highlight Text on Focus

Highlights the field when selected.

Use Enter to

Allows to use the “Enter” key to pass the focus to the next field.

Field with Initial Focus

Determines the field starts with focus when accessing the application. This option doesn’t work with fields that contain a watermark.

Highlight Field with Error

Focus the field with the error when submitting the form.

Use a template from the HTML Editor

Allows to use the TinyMCE editor. You can edit and create your HTML Templates.

Related Links

Related Videos ▷

Control Edit fields

This interface is useful for editing the field settings and their position to display.

Edit Fields

1 Fields	Label	Datatype	New	Update	Read-only	Required	PK	DB value (Insert)	DB value (Update)
5 PAGE: PAG1	2	3					4		
6 BLOCK: FORM_CUSTOMERS									
customerid	Customerid	Text	☑	☑	☑	☑	☑		
companyname	Companyname	Text	☑	☑	☐	☐	☐		
contactname	Contactname	Text	☑	☑	☐	☐	☐		
contacttitle	Contacttitle	Text	☑	☑	☐	☐	☐		
birthdate	Birthdate	Date	☑	☑	☐	☐	☐		
country	Country	Text	☑	☑	☐	☐	☐		
7 PAGE: FIELDS NOT SHOWN									
regionid	Regionid	Integer	☐	☐	☐	☐	☐		
stateid	Stateid	Select	☐	☐	☐	☐	☐		
city	City	Text	☐	☐	☐	☐	☐		
address	Address	Text	☐	☐	☐	☐	☐		

Edit Fields Configuration.

Fields

It allows accessing the field settings (a pencil icon on the left). You can change the field position by dragging them to the desired position. Drag a field to “fields not displayed” if you don’t want it in the app.

Label

It defines the title of the field in the app. For example: if the field name in the database is fld_txt_customer_name, you can display the label “Customer Name”.

Data type

It informs the data type of the field.

New

It defines if the field is available when inserting new records.

Update

It defines if the field is available when updating records.

Read-Only

It defines the field as a label. The user can’t change its value.

Required

It defines if the field must contain a value.

PK

It defines the Primary Keys fields.

DB value (Insert)

Defines a default value for the field when inserting a new record, like an auto-increment, Date, DateTime, or IP.

DB value (Update)

Defines a default value for the field when updating a record, like an auto-increment, Date, DateTime, or IP.

Page

It shows the pages available in the application. All apps have a page, at least. Each page contains one or more blocks.

Blocks

It shows the blocks available in the application. Blocks contain fields. All apps have a block, at least. A block is displayed if it contains one or more fields.

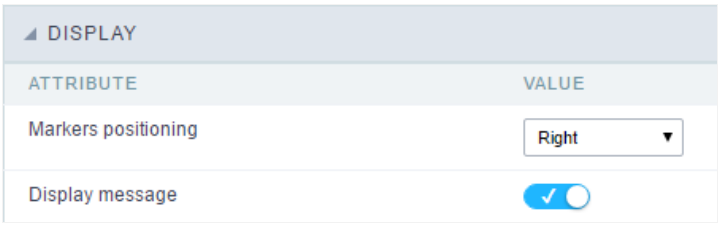
Page Fields Not Shown

Here we can see the fields that are not in the application.

Observe that you can drag any line to the desired position, blocks, and page. Pages contain Blocks, and Blocks contain fields.

Display Settings

Settings of messages display.



Display Interface.

Attributes #### Markers positioning Set the position of the markers that Indicates required fields.

Display message
Set it if you want to display the message of the required field.

Related Links 🔗

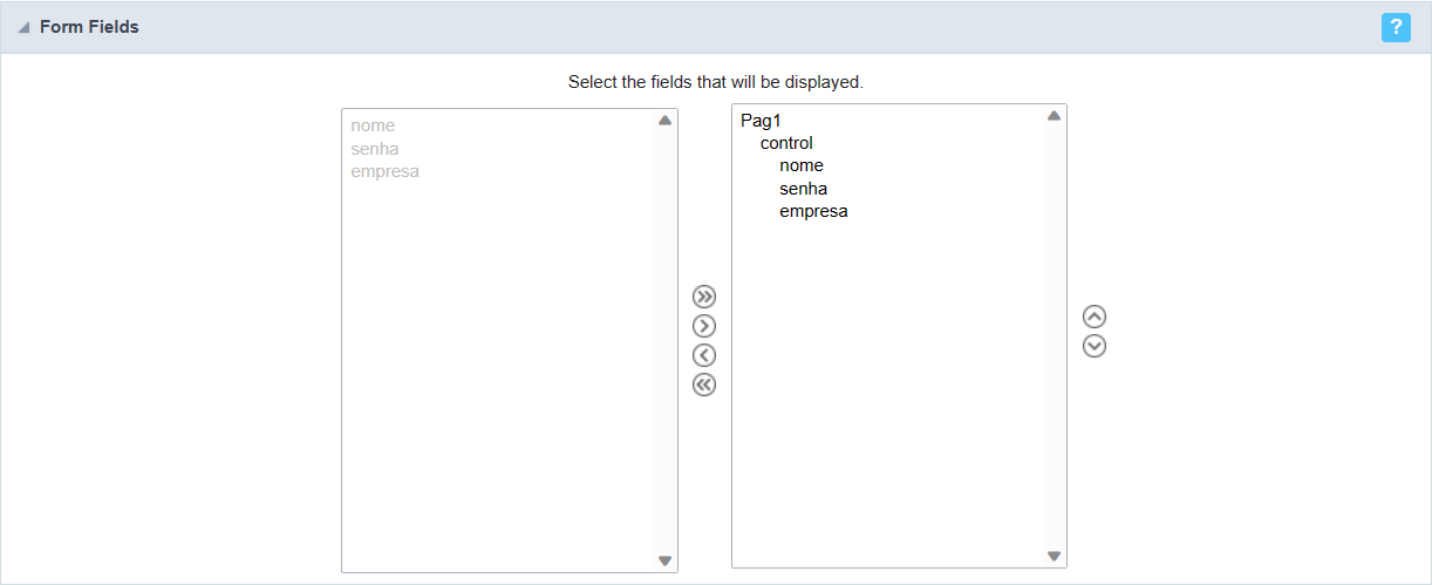
Related Videos ▷

Control Fields positioning

In this configuration screen we have a list of all the fields available in the application, whether they are fields mapped from the table or virtual fields (Created only in the Scriptcase interface).

It is also possible to allow the end user to manipulate the application's fields in the way they prefer, for that we must add the **columns button** in the application's toolbar.

In the **left column**, we have a list with all the fields available for use and in the **right column** we have a list of the fields that were selected to be displayed in the running application.



Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

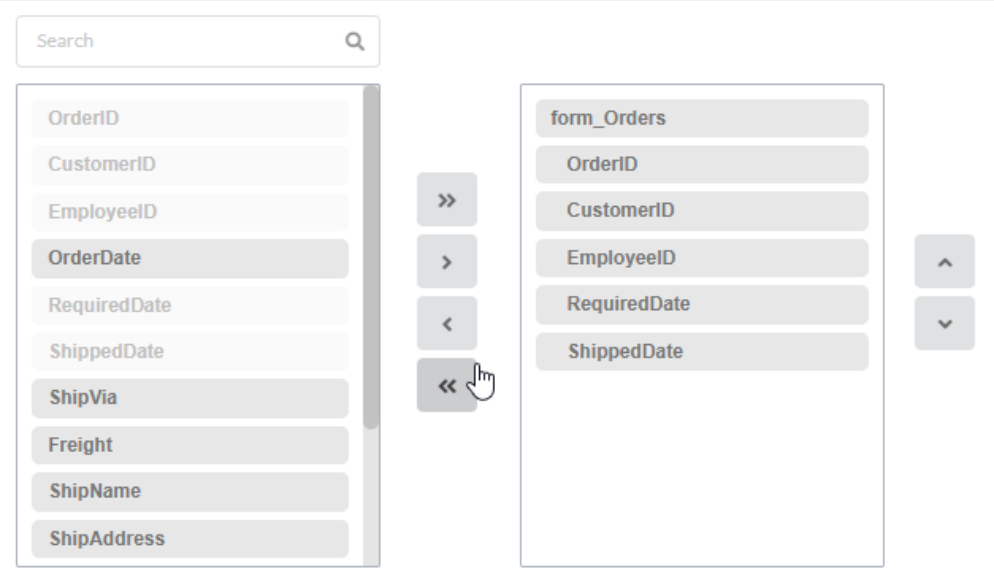
- » - Move all fields to the right.
- › - Moves only selected fields to the right
- ‹ - Moves only the selected fields to the left.
- « - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the › button.

To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning



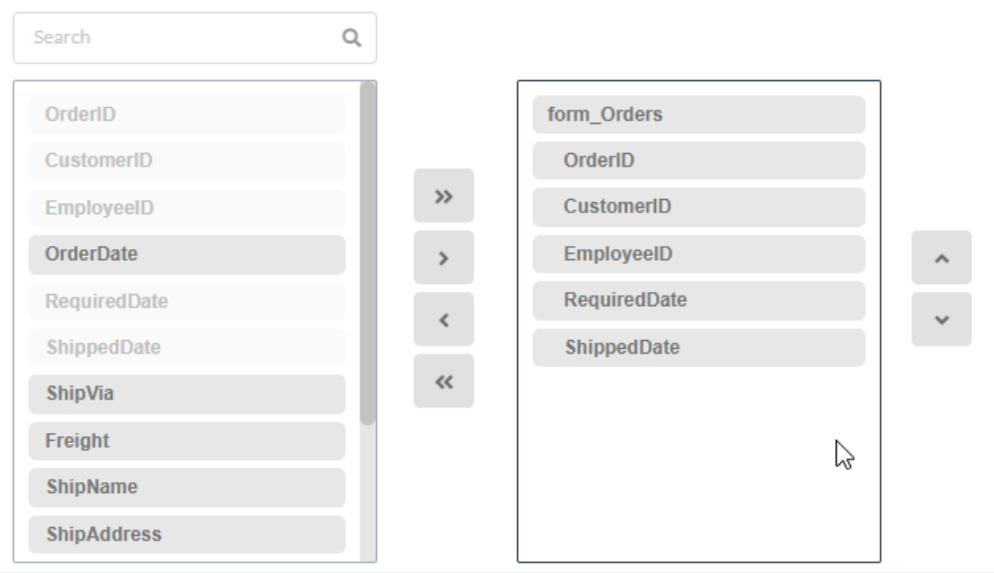
Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

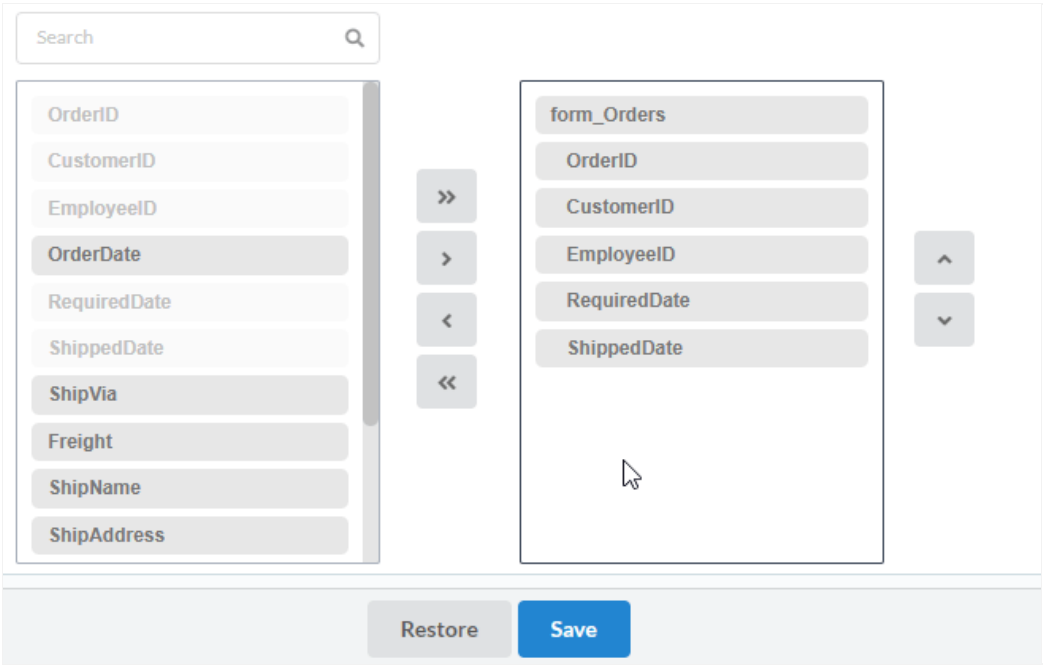


Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.



Related Links

Related Videos

Control Toolbar

The application toolbar has two segments: Top and Bottom, in a way that is possible to define to display buttons into both areas. Those areas work independently, allowing them to display the same button, for example.

It's also possible to select the buttons and their position if the application is running on a mobile device.

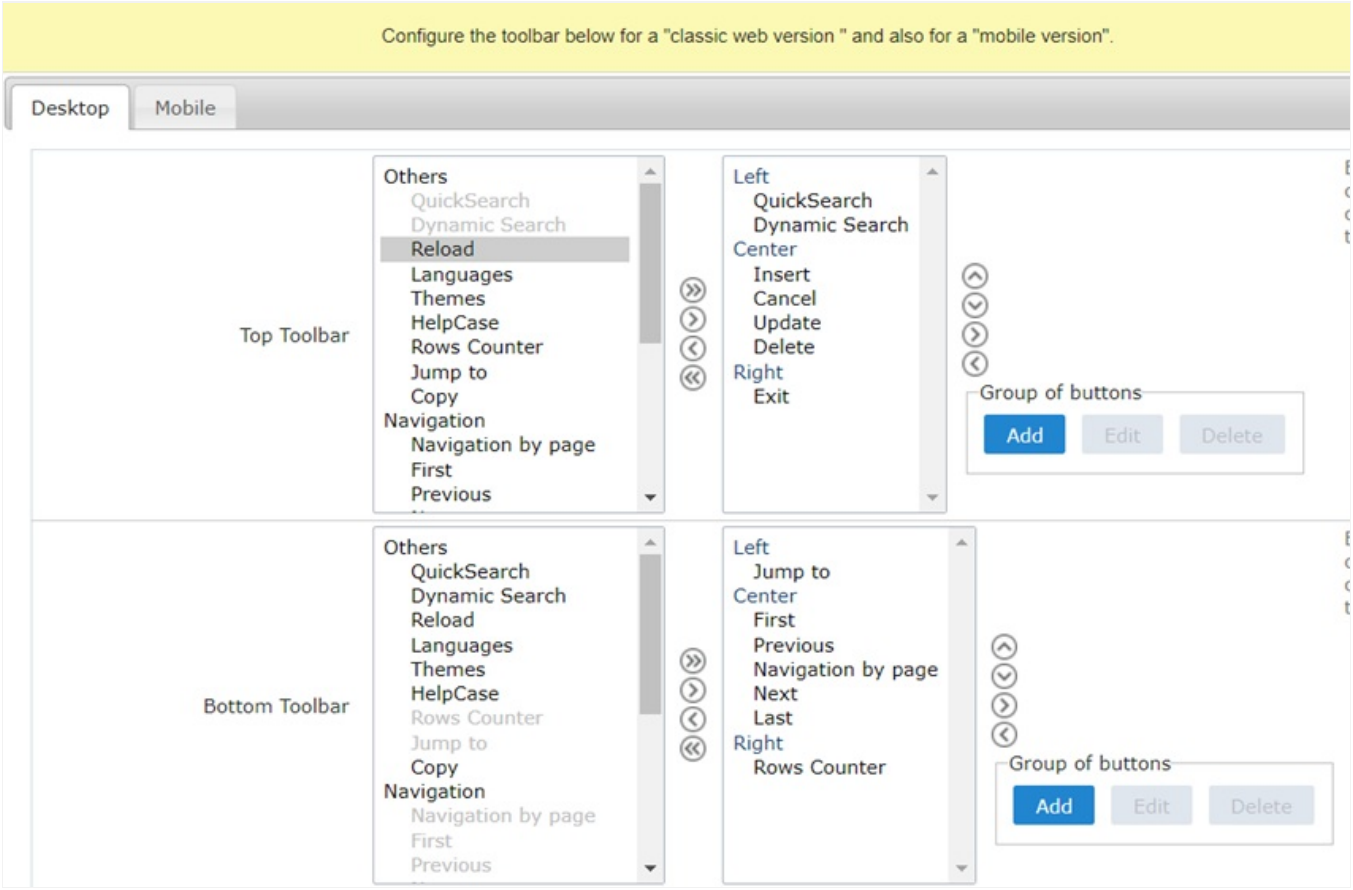
Toolbar

Desktop

Here we must inform the toolbar settings for the "Classic Web Version" mode and which buttons are available in the application when accessed from a **Desktop** environment.

Mobile

Here we must inform the toolbar settings for the "Mobile Version" mode. That is which buttons are available in the application when accessed from a **Mobile** dispositive.



Navigation:

Buttons relative to the navigation of the application.

Next	Move to the next page that can be a single record or a list of records.
Previous	Returns to displays the previous page records or a single record.
First	Move to the First page or record
Last	Move to the Last page or record
Exit	Close the application
Navigation by page	Displays a “page-number” navigation bar. **Example:** 1 2 3 4 5
Reload	Displays a button to reload the query data

Export:

The options available to export the Records. Scriptcase generates the following export formats for Forms:

PDF	Generates all the data of the application in a PDF format.
Print	Creates an HTML with the records ready for printing.

Update:

The CRUD options available in the Form.

Insert	Inserts the record into the database.
Update	Saves the changes made in a record.
Delete	Deletes the selected record.
Cancel	Cancel the changes made in a record before the insertion.

Others:

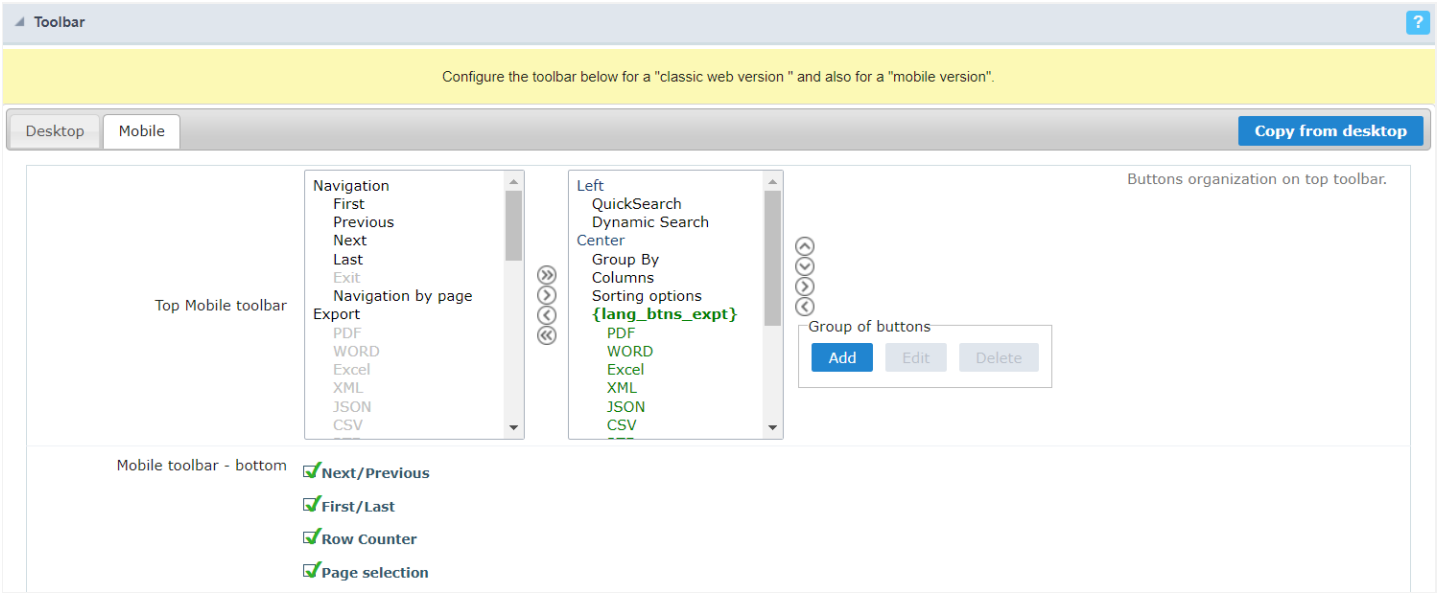
Other options available in the Form application.

Jump To	Move to the informed page or record.
Copy	Copy the current record data to another one.
Quick Search	Perform a quick search in the records of the application.
Dynamic Search	It displays the fields of the search to filter the records.
Languages	Displays a Combobox with the languages available in the project properties.
Themes	Displays a Combobox with the languages available in the project properties.
Rows Counter	Displays the number of records retrieved in the application.
HelpCase	Displays a button to open the help page.

Separator:

————————	Displays a line separating the buttons.

Toolbar Mobile



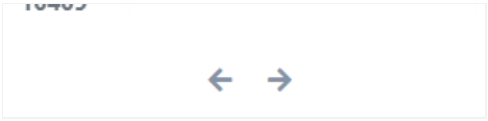
Top Mobile toolbar

It has the same options as the **Desktop** version, adding only the item “Copy from desktop”, which, when clicked, makes a copy of the items from the upper toolbar of **Desktop** to **Mobile**.

Mobile toolbar - bottom

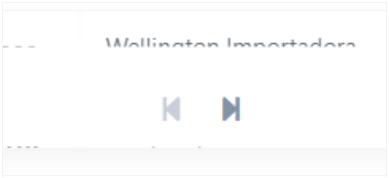
Next/Previous

Enables navigation to the next and previous page on mobile devices.



First/Last

Enables first and last page navigation on mobile devices.



Row Counter

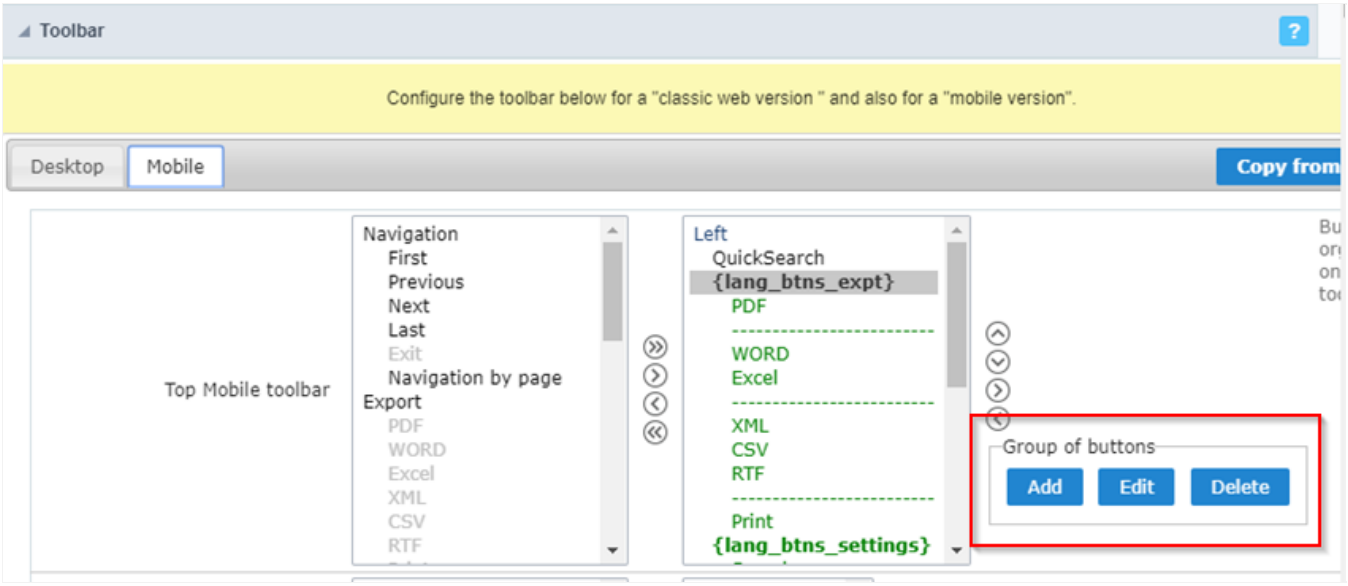
Enables the record counter showing the application's total records

Page selection

Enables page navigation on mobile.

Bottons Group

The **Group** option allows you to group a set of buttons of the application toolbar to display them as a dropdown, for example.



Add

Add a new group of buttons.

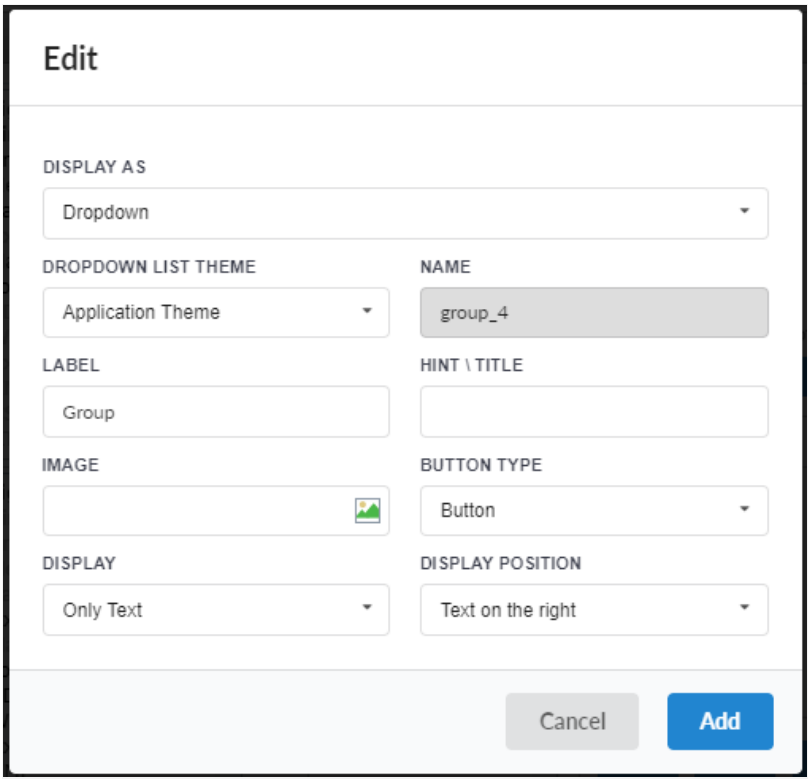
Edit

Edit an existent group of buttons.

Delete

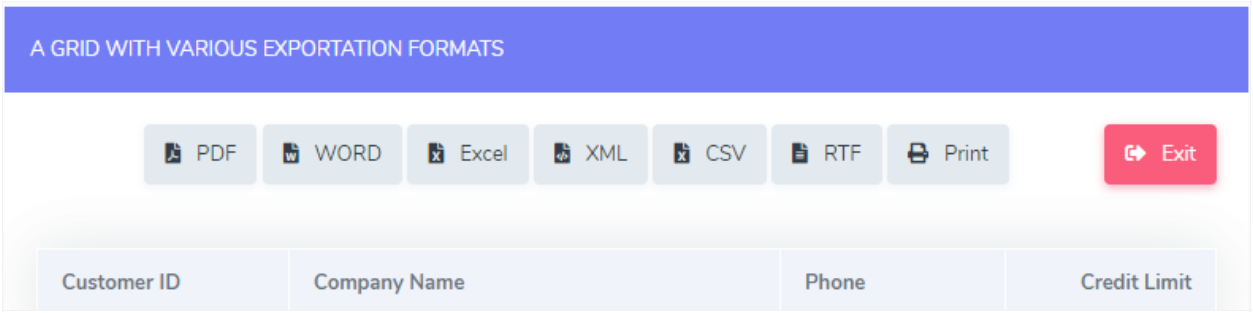
Delete the selected group of buttons.

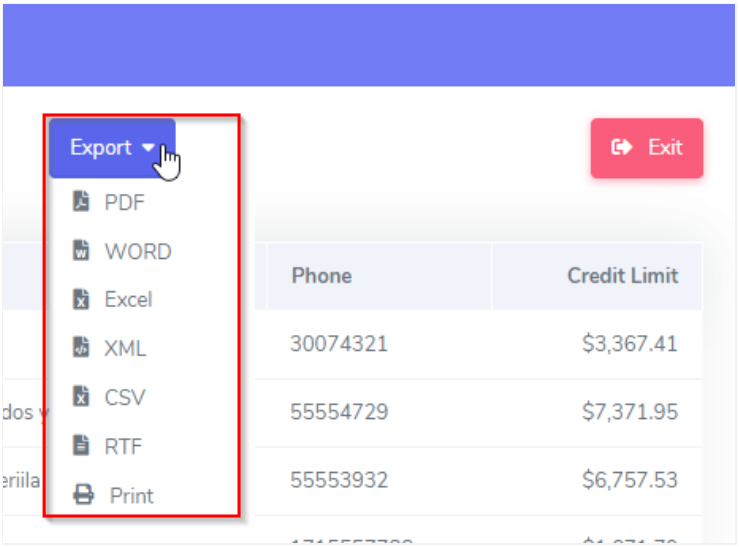
When you press the **Add** or **Edit** option, you can see the settings to configure the grouper:

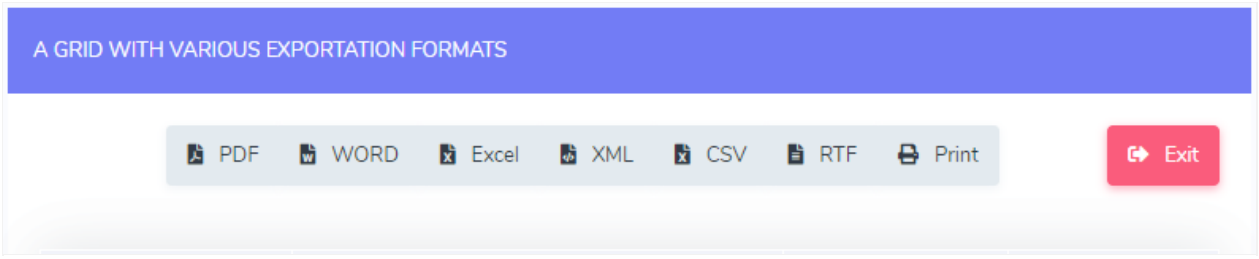


Display As

Allows displaying the group button as **Dropdown** or **Side by Side**.







Dropdown List Theme

Allows defining the Dropdown theme selecting between **Application theme** and **Button theme**.

Name

Allows defining a name for the button group.

Label

It is the displayed name for the button group in the application.

Hint\Title

Displays a hint to the end-user when the mouse is on the group of buttons.

Button Type

Allows displaying the button group as a Button, Image, or Link.

Image

Allows selecting an image for the button.

Display

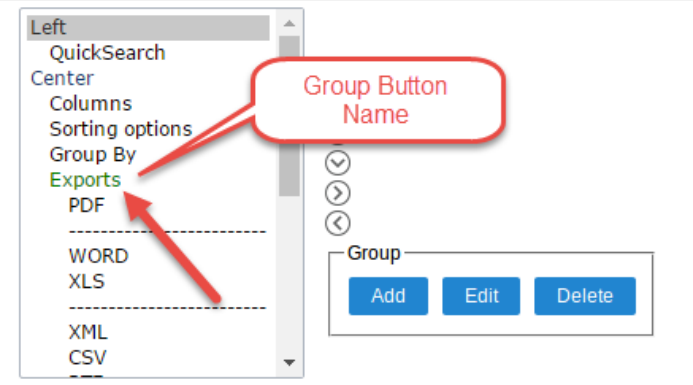
Defines if the button displays only Text, only image, or both.

Display Position

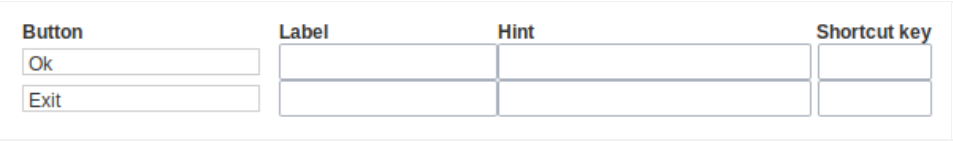
Defines the position of the Text or Image (Text to the right, Image to the right).

After creating a button group, you need to move the grouped buttons below of the Button Group and then move them to the

right. Like the image below:



Buttons Settings



Button:

It displays the buttons available in the application.

Label:

Allows defining the buttons name that to display for the users.

Hint:

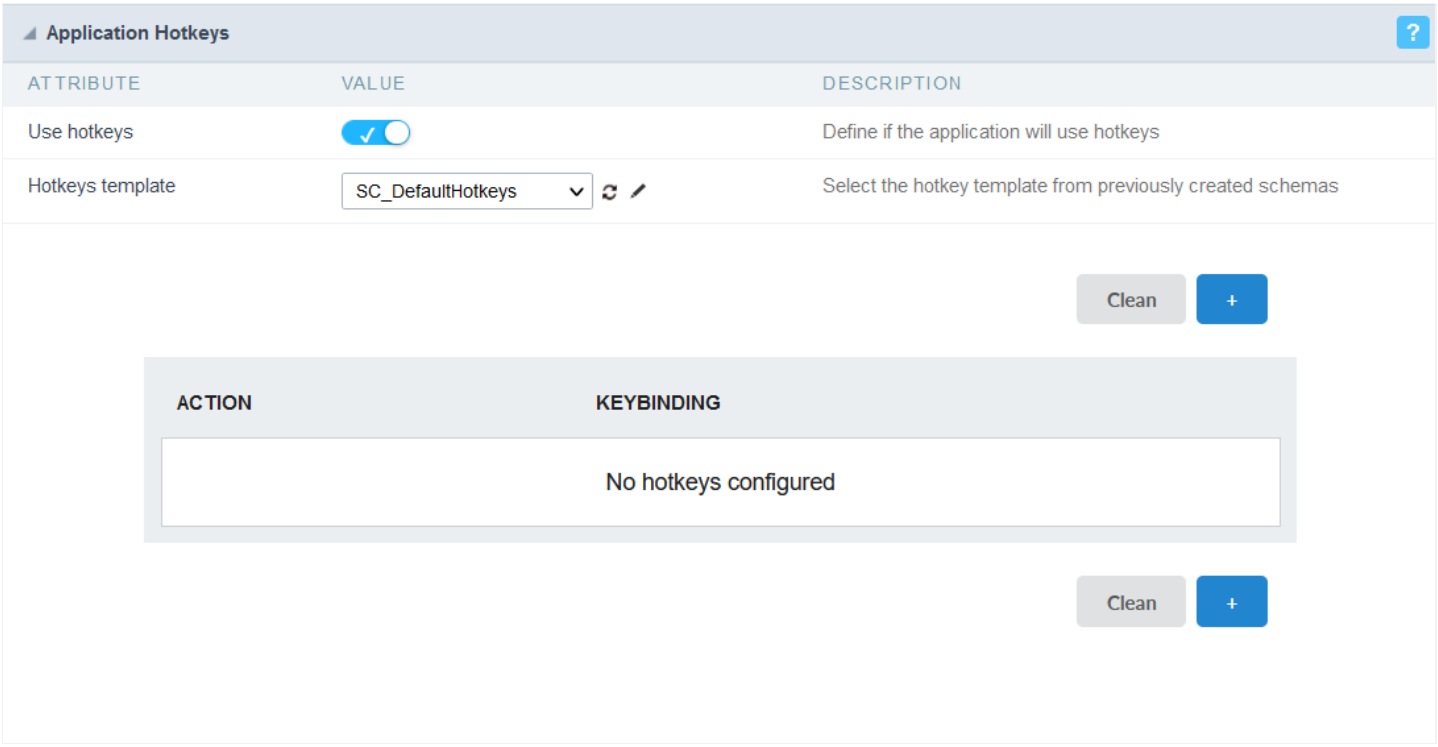
Allows defining the buttons hint that to display for the users.

Shortcut Key:

Allows defining the shortcut key for each button.

Application Hotkeys

Scriptcase allows creating shortcut keys to your applications. You can select a predefined template or create specific actions for an application.



Use hotkeys

Defines if the application uses hotkeys. When you enable this option, the default shortcut keys settings are disabled.

Hotkeys templat

Select the hotkey template previously created.

Action

Selects the triggered action when pressing the selected key.

Keybinding

Selects the keys responsible for executing the chosen action.

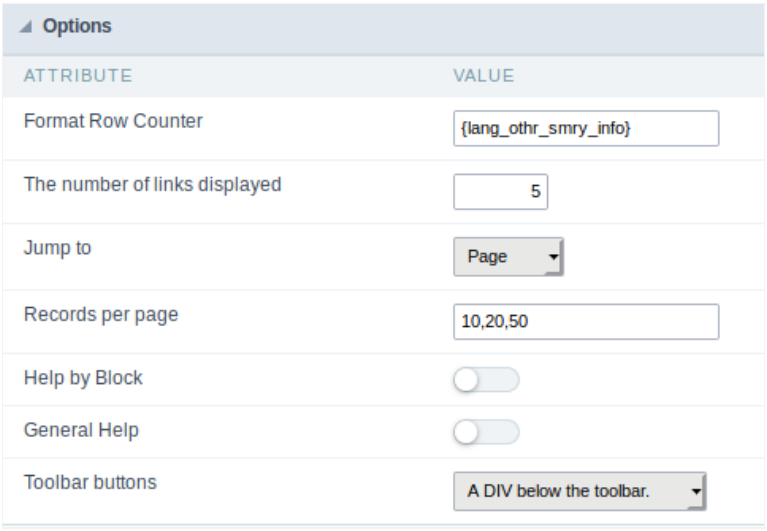
Add “+”

Adds a new action on the keys list.

Clear

It clears the selected hotkeys preference.

Options



Rows Counter:

Allows to display the Rows Counter or not.

Example: (1 to 10 of 200)

The Number of Links Displayed:

Defines the number of links per page, when the navigation option is disabled.

Jump To:

Move to the informed record page.

Records by page:

Allows defining the number of records to be exhibited for each page. To show all records use the option “all”.

Example: 10,20,30, all

Help by Block:

Indicates if the helps messages from relatives fields are grouped by block. In each field, we can define a help text. With this option activated, it shows up an icon in the block title bar to call the help page.

General Help:

The General Help “consolidates” all the fields help pages in a single page, putting an icon in the toolbar to call the help page.

Toolbar Buttons:

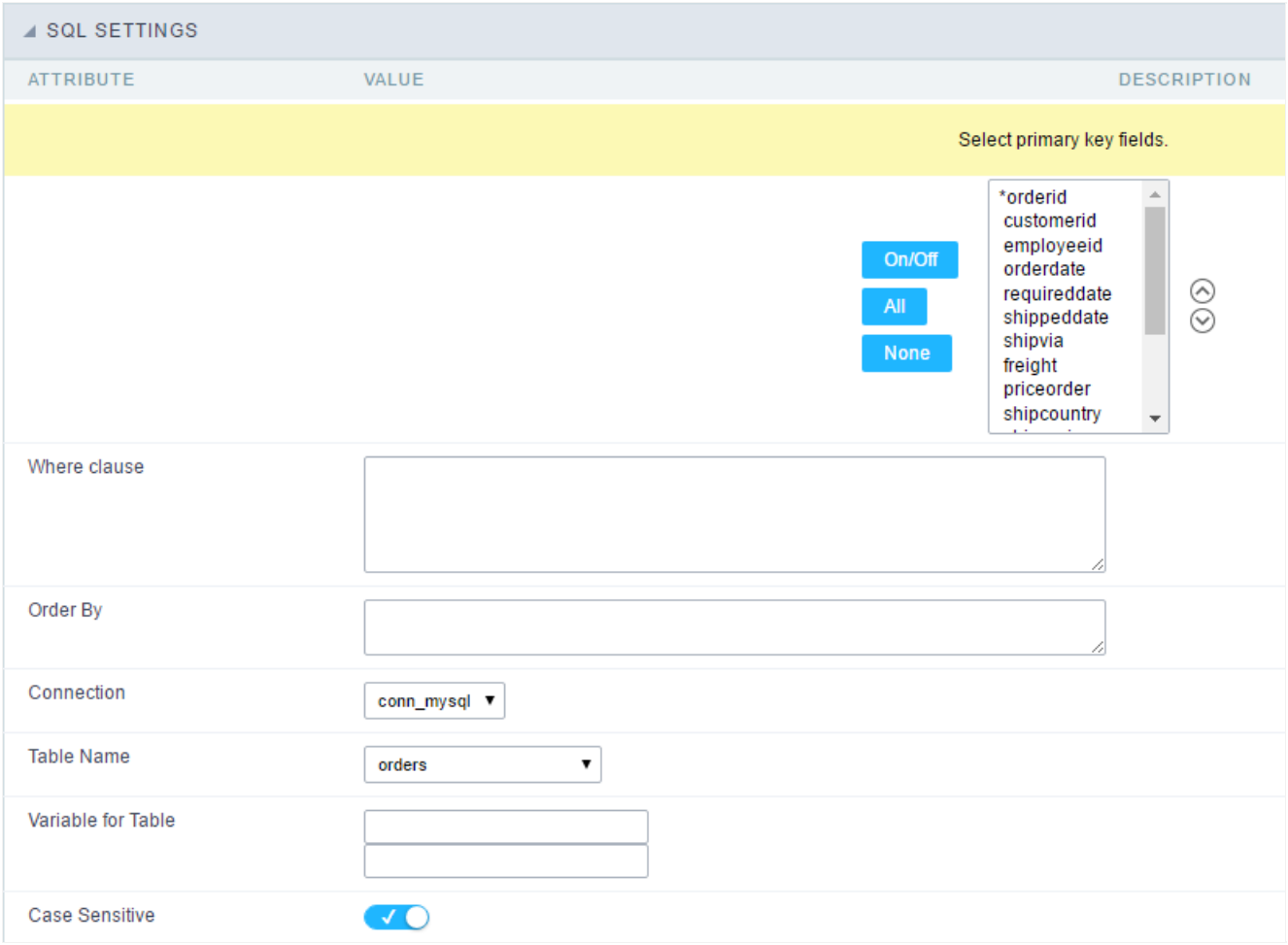
The type of view for the Toolbar buttons (A DIV below the toolbar, Modal).

Related Links 🔗

Related Videos ▷

Control SQL Settings

This interface allows configuring the related database settings, such as the Primary Key, Filters, Sorts.



SQL configuration Interface.

Select primary key fields

It lets you define the Primary key of the Form. ScriptCase already identifies Primary Keys, but you can manually inform or change it by using the buttons beside the fields list. See how the buttons work:

- **On/Off** : Adds or Removes the attribute that defines the primary key for the field. The primary key fields have an asterisk beside their names.
- **All** : Defines all fields as Primary Keys.
- **None** : Defines none fields as Primary Keys.
- **Sorting Button** : These are the arrows on the right side of the Combo box. It allows ordering the fields of the Primary Key, placing it in the desired order. To order them, click on the field and use the arrows to move it.

Where clause

It allows adding a WHERE clause to filter the SQL records.

Order By

It allows adding an ORDER BY clause to determine the order to display the records. By default, it uses the primary key to sort the records.

Connection

It allows defining the database connection of the application. You can change the connection to another one that has the same table.

Table Name

It informs the database table used in the Form.

Variable for Table

It allows to use a variable to change a part of the string containing the table name.

Variable for Table Configuration.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the table you want to replace (replace from).

Case sensitive

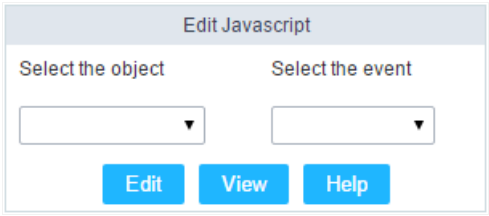
It defines if the database connection uses case sensitive or not.

Related Links

Related Videos

JavaScript

To coding with JavaScript in Scriptcase Form application, we must associate a JavaScript event to a form field.



Edit JavaScript Interface.

Select the object

This Combo box displays the fields of the form application, and also the form itself as an object

Select the event

Use it to associate the event with the selected field to apply the JS code. View the events available:

- **onclick**: Acts when clicking on the field.
- **onblur**: Acts when the focus leaves the object.
- **onChange**: Acts when the focus leaves the object, and there are changes in the value.
- **onFocus**: Runs when the focus gets in the object.
- **onMouseOver**: Runs when the mouse pointer hovers the object.
- **onMouseOut**: Runs when the mouse pointer moves out the object.

Events related to the Form

The events below are associated directly with the form object.

- **OnLoad**: This event runs when the page is loaded, also when clicking on the navigating buttons.
- **onSubmit**: This event runs when clicking on the “New”, “Save”, and “Delete” buttons.

Edit JavaScript

*Select the object and event, then click on the Edit button. It opens a page to inform custom JavaScript routines and standar

![Edit JavaScript Interface][javascript_edicao_code]
Edit JavaScript Interface

> The JavaScript language doesn't have the same behavior in all the browsers available. A tip is to test running the applicatic

- **OnClick Example**
  - When clicking on a field of type radio, you can enable or disable form fields according to the selected value.

```
if(document.F1.gender[0].checked){
   document.F1.maternity.disabled = false;
   document.F1.maternity.style.background='FFFFFF'
}
if(document.F1.gender[1].checked){
  document.F1.maternity.disabled = true;
  document.F1.maternity.style.background='FCEEBC'
}
```

To access the values of a radio field, you need to use the index.

- **OnBlur Example**

- You can define a warning for the field “weekly_work_time” when the focus is getting out it.

```
if (document.F1.tp_point[0].checked  && document.F1.weekly_work_time.value > '20')
{
    alert("The work time exceeds the limit allowed");
    document.F1.weekly_work_time.value = "";
    document.F1.weekly_work_time.focus();
}
```

- **OnChange Example**
  - By modifying the “Salary” of an employee and leaving the field, we’ll check if his “position” is “gardener”.

```
if (document.F1.salary.value > 2000.00 && document.F1.position.value == 'gardener'){

alert('When I grow up, I want to be a gardener);
}
```

- **OnFocus Example**
  - After informing the purchase value and selecting the payment method in a Select object “Select: pay_method”, the JavaScript code below calculates the value of the purchase.

```
if (document.F1.pay_method[document.F1.pay_method.selectedIndex].text == 'Money')
{
     document.F1.total.value = document.F1.pauchase_value.value;
}
```

- **OnMouseOver Example**
  - You can change the style (background color, font and font color) when the mouse hovers the field.

```
document.F1.field_name.style.bgColor = "0FFCCA"
```

- **onMouseOut Example**
  - Sets the background color when the mouse moves out from the field.

```
document.F1.field_name.style.bgColor = "FFFFFF"
```

Related Links 🔗

Related Videos ▷

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application.

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Enable CSRF

With this option enabled, the scriptcase prevents a malicious attack on a page where unauthorized commands are transmitted through a user the page trusts.

These attacks are known as a “Cross-Site Request Forgery” attack.

Related Videos ▷

- [nguage selection
- Contact Form template
- Registration form template
- Control introduction
- Responsive login template
- Control overview
- Form overview

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Headers

Disable XSS Auditor

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Captcha

Captcha

Use Captcha

Defines if the application uses Captcha.

Number of Characters

Amount of characters in the Captcha image.

Character List

List of characters used in the Captcha.

Label

The message displayed for the Captcha.

Error message

Captcha error message.

Height

Height of the Captha image. (in pixels)

Width

Width of the Captha image. (in pixels)

Font Size

Font Size of the Captha image. (in pixels)

Reload

Display the refresh button in the Captcha.

Select one layout

It offers layouts for display the Captcha.

Recaptcha

ReCAPTCHA is an API provided by Google for forms. It adds security, preventing automatic submission of forms through robots.

reCAPTCHA sample:

1. First, we must request an API Key to activate reCAPTCHA into a Scriptcase application by following the steps below:

   To get a **Site key** and **Secret Key** go to the link: https://www.google.com/recaptcha/admin#list.
   See the image:

Label

It is a project identifier to create the reCAPTCHA keys.

Choose the type of reCaptcha

We must choose the option **reCAPTCHA V2**.

Domains

We can insert multiple domains (one per line) to limit the API uses.

1. Then, we need to accept the Terms of Service (“Accept the reCAPTCHA Terms of Service”).
2. When clicking on **Register**, the page refreshes and shows the integration of reCAPTCHA information. There we can get the **Site Key** and **Secret Key**:

1. Now, we can set the Scriptcase application security:

Site Key

Key generated by Google after reCAPTCHA project creation.

Secret Key

Key generated by Google after reCAPTCHA project creation.

Theme

Defines the reCAPTCHA color. There are two options:

- **Light** :
- **Dark** :

Type

The type of reCAPTCHA. There are two options:

- **Audio**:
- **Image**:

Size

The size of reCAPTCHA. There are two options:

- **Normal** :
- **Compact**:

Position

Here we can define the reCAPTCHA component alignment:

- **Left**: Position the reCAPTCHA component at the left.
- **Center**: Position the reCAPTCHA component at the center.
- **Right** : Position the reCAPTCHA component at the right.

Related Links

- Login with language selection
- n template
- form template

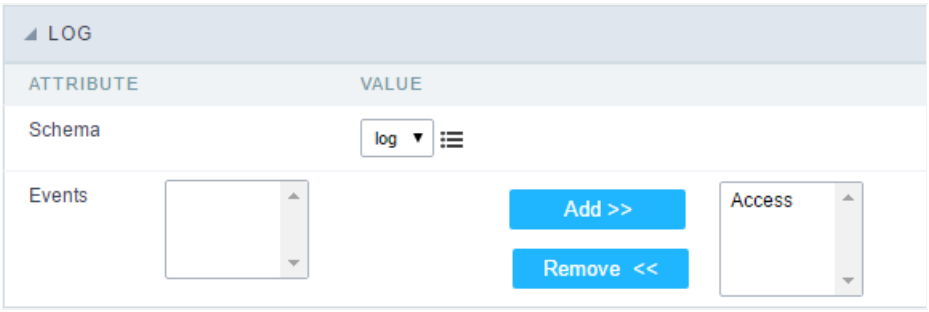
Related Videos

- Responsive control template
- Control overview
- Form overview

Control Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

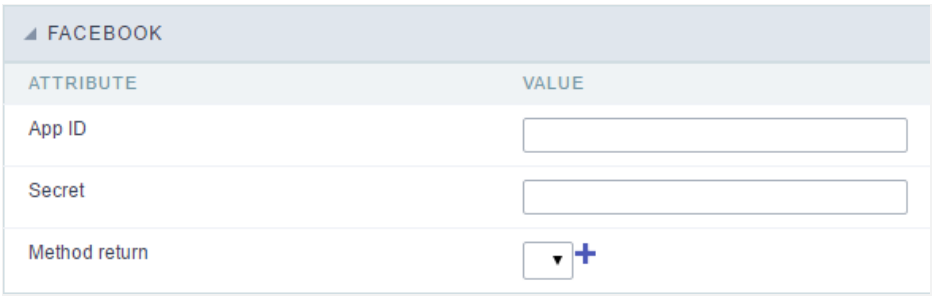
Related Links

Related Videos

Authentications

Scriptcase can link its applications using OAuth methods. These methods allow you to integrate a Scriptcase project with the following data sources:

Facebook



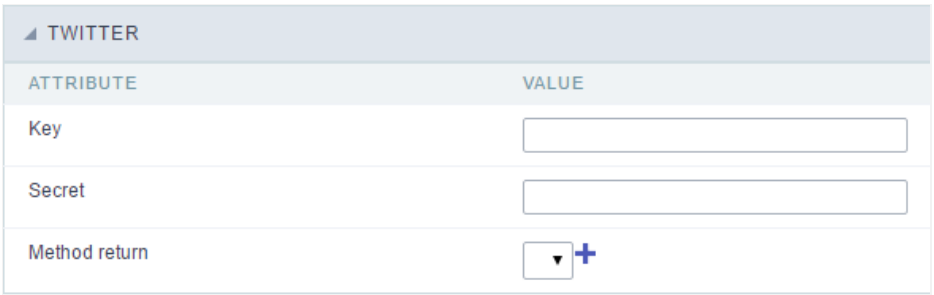
Facebook fields authentication interface.

When you create a button Facebook authentication, you must fill in the fields:

- **App ID** : Facebook application ID. This ID will be provided by Facebook Developer's page.
- **Secret** : Secret key provided by Facebook, so the application can be executed.
- **Method return** : Method that will be performed when Facebook service returns the requested data. You will have to create that method according to our validation rules, using the option "Programming > PHP Methods"

For building applications, visit: Facebook app development

Twitter



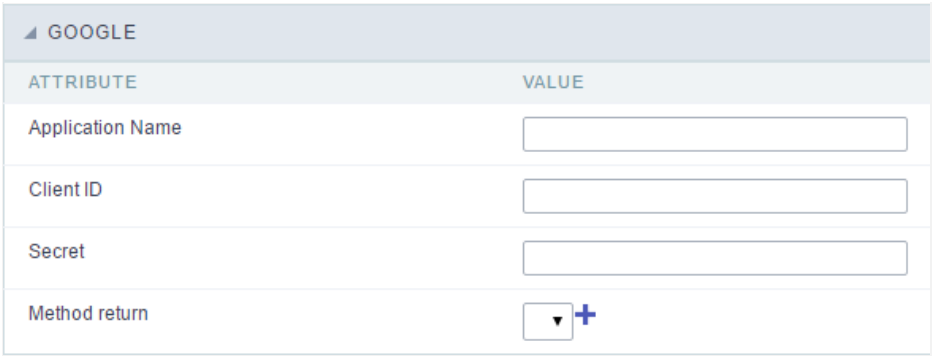
Twitter fields authentication interface.

When you create a button "Twitter Authentication", you must fill in the fields:

- **Key** : Twitter application ID. This ID will be provided by Twitter Developer's page.
- **Secret** : Secret key provided by Twitter, so the application can be executed.
- **Method return** : Method that will be performed when Twitter service returns the requested data. You will have to create that method according to our validation rules, using the option "Programming > PHP Methods"

For building applications, visit: Twitter app development

Google

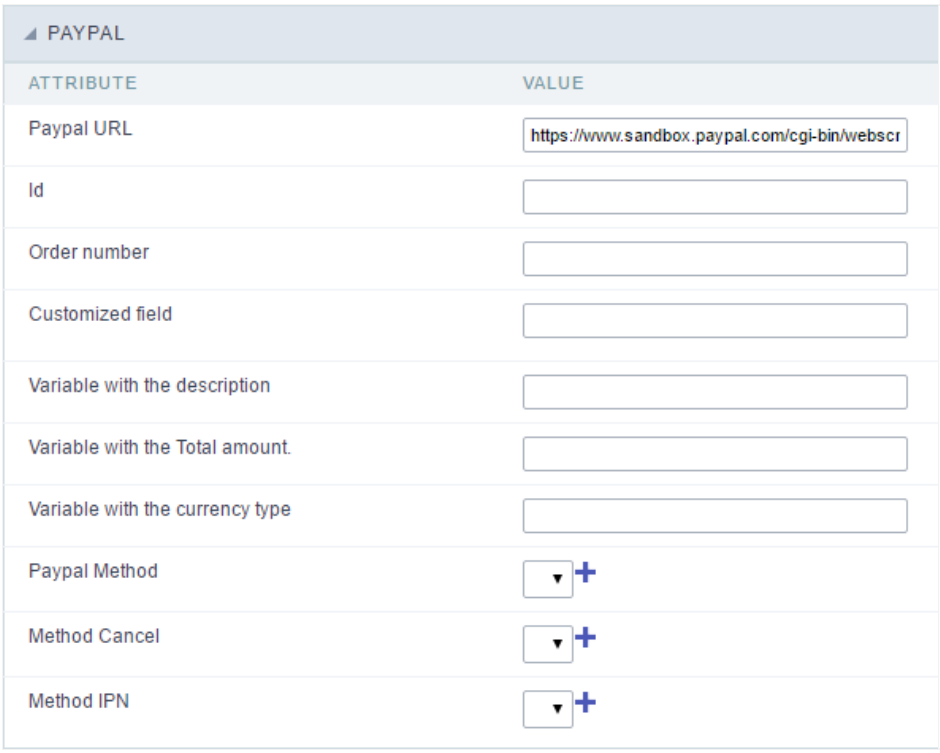


Google fields authentication interface.

- **Application name** : Name of the application created on Google+.
- **Client ID** : Google+ application ID. This ID will be provides by Google Developer's page.
- **Secret** : Secret key provided by Google+, so the application can be executed.
- **Method return** : Method that will be performed when Google+ service returns the requested data. You will have to create that method according to our validation rules, using the option "Programming > PHP Methods"

For building applications, visit: Google+ app development

PayPal



PayPal fields authentication interface.

To create the integration with PayPal, you must fill in these fields:

- **PayPal URL** : This URL can be to the PayPal SandBox or to the real PayPal API. The SandBox is used to perform tests with the application before applying the integration in a real environment.
- **Id** : PayPal ID to receive the sales money. Usually the registered email as business.
- **Order number** : Order Number to be stored in PayPal. Also used to treat the receipt of the IPN in order to identify and process the order as paid.
- **Customized field** :Custom field to be stored in PayPal. Also used to treat the receipt of the IPN in order to identify and process the order as paid, if you want to do some additional checking.
- **Variable with the description** : Using this field you can place the Global variable that will contain the description to be sent to PayPal.
- **Variable with the Total amount.** : Method to be ran when PayPal service return the data requested.

- **Variable with the currency type** : Use to specify the payment currency. If the currency variable is not included, the currency defaults to USD.
- **PayPal Method** : Method that will be called after the User complete the PayPal transaction. You will have to create that method according to our validation rules, using the option “Programming > PHP Methods”
- **Method Cancel** : Method to be executed if the user cancels the PayPal transaction. You will have to create that method according to our validation rules, using the option “Programming > PHP Methods”
- **Method IPN** : Method that will be called by web service from PayPal to confirm the payment of a transaction. This method is called by web service and it has no HTML output to the end-user. You will have to create that method according to our validation rules, using the option “Programming > PHP Methods”

For PayPal configuration, please visit: Configuring PayPal credentials

The variables list returned by the PayPal API can be found below:

[mc_gross]
[invoice]
[protection_eligibility]
[address_status]
[payer_id]
[tax]
[address_street]
[payment_date]
[payment_status]
[charset]
[address_zip]
[first_name]
[mc_fee]
[address_country_code]
[address_name]
[notify_version]
[custom]
[payer_status]
[business]
[address_country]
[address_city]
[quantity]
[verify_sign]
[payer_email]
[txn_id]
[payment_type]
[last_name]
[address_state]
[receiver_email]
[payment_fee]
[receiver_id]
[txn_type]
[item_name]
[mc_currency]
[item_number]
[residence_country]
[test_ipn]
[handling_amount]

[transaction_subject]

[payment_gross]
[shipping]
[ipn_track_id]

Related Links 🔗

Related Videos ▷

Control - Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid	✓	✓
Procedure	✓	✓
Chart	✓	✓
Form	✓	✓
Control	✓	✓
Calendar	✓	✓
Search	✓	✓
PDF Report	✓	✓

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven't manage fields.

Related Links

Related Videos ▷

- [nguage selection
- Contact Form template
- Registration form template
- Control introduction
- Responsive login template
- Control overview
- Form overview

Control - Text Field

General Settings

This type of field allows the developer to create quickly inputs to insert and update data, where the final user can inform its data to be allocated in its database.

Type of Data

Defines the type of the field in the application. In this case we should select **Text**.

Label

Defines the title that will be displayed in the field when the application is executed. The terminology used in the interface is fundamental to the system have a good usability, whe should use names and familiar terms to the final user of the application, instead of using terms from the system.

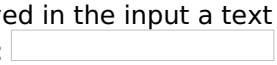
For example, this text field that has the name **customerid**, the client would have a much better understanding of the functionality of the field when we define the label as **Customer Name**.

Not only a fixed text, the **Label** attribute allows the use of langs to define the field's title, making it possible to use your application in a multi language project.

Label below field

Defines the message to be displayed below the field.

Watermark

Informing a text to the **Watermark** it will be displayed in the input a text as an example that can be informed in the field. The result after set will be this: 

Initial Value(type)

Allows the initial definition to the field when the form is in insert mode. It is possible to choose between two options:

Defined Value: When this option is selected, the Initial Value attribute will be available, where we should inform the field's initial value. For example, my initial value is **Arlindo**, when a new register is inserted, the field **Seller Name** will be initialized as Arlindo.

System Data: When this option is selected, the initial value will be the actual date of your computer's system.

Amount of Characters

Allows to set the width of the text field's input that varies with the amount of characters informed. Although, if the amount of characters typed are greater than the setting, the text will be pushed to the left, to keep the maximum amount of characters as defined.

Show HTML Content

When this option is active every HTML, CSS and JavaScript content that are in the database will be displayed with the main value.

Validation Image

When this option is active, a image will be displayed next to the informed field if the field is according to the settings defined by the developer.

In the example below, the field was set to receive at least 5 characters, see what happens when informed

only 4 characters:

However if informed 5 or more characters the field will be displayed as:

Password Field

When this option is active, the text field will be converted to the format used in password fields. For example:

View password characters

By enabling this option, a button will be displayed in the password field so that the password is displayed when clicked.

Save Variable

Allows to save a session variable(global variable) with the field's value, to be used in others applications.

For example, in the login form the username can be saved in session and displayed on the header of others applications.

Variable Name

In this attribute we should define the name of the session variable, active in the previous item, that will receive the field's value.

We should inform only the variable's name, - **var_rating**.

The method to use its value is global variable.

Field Mask

Defines the field mask. There are two types of mask described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

For example, it is possible to set the mast to display a telephone number:

It will be show with this format on runtime:

It is also possible to set the field mask like those examples:

Field mask examples:

Telephone number

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678

Field	Mask Input	Typed Value	Formatted Value
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Validate with mask

When this option is active it is possible to insert the data with the mask set in the **Field Mask** option.

Field size in database

Defines the field's size related to the size set in the database. This value is already set automatically by default when the application is generated.

Hidden Field

When this option is active, the field will be hidden in the application on runtime.

Label Field

When this option is active, the field will be altered to only a label where the info will be displayed, where updates or inserts will not be possible.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disable Field

Allows the developer to disable the field, so the user can not type a value according with the option defined by the developer.

The available options are: **No** - Initial value of the attribute, this option does not disable the field.
Update Mode - This option only disables the field when editing existing registers.
Insert Mode - This option only disables the field when inserting new registers.
Insert / Update Mode - This option disables the field when editing exciting registers or when inserting new registers.

HTML Type

HTML object used to display the field in the form.

SQL Type

Informs the type of the field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed.

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template

Related Videos

- Responsive login template
- Control overview
- Form overview

Control - Multiple Lines Text Field

General Settings

Multiple Lines Text field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Multiple Lines Text , you can inform a Text value to the field in multiple lines.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.

- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Define the amount of characters allowed for the field.
- **Validation Image** : Allows to display an image next to the field when it is being validated.

- **Lines** : Allows to define the amount of lines that the field will have at start.

- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the field size in bytes. It is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed.

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.

- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Integer Field

General Settings

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Label:

It sets the title that will be displayed in the field when you run the application. The terminology used in the interface is fundamental for your system has a good usability. In this case, we recommend you to use names and terms familiar to the end user of the application, instead of using terms originated in the system.

Label below field

Defines the message to be displayed below the field.

Watermark

Informing a text for the **watermark** it will be displayed on the input text with an example of what can be reported in the field. The result after that will be configured:

Initial value (type)

Allows setting an initial value for the field when the application is in inclusion. You can choose between two options:

Pre-set value: When you select this option, the Initial Value attribute will be made available in the place where we inform the initial value of the field.
For example, my initial value is **Arlindo** when inserting a new record the **seller_name** field will start with Arlindo.

System date: When you select this option, the initial value will be the current system date of your computer.

> If you select the type system date, it will be not necessary to fill in the initial value attribute.

Amount of characters

It allows you to set the width of the input text field according to the amount of characters. However, if the quantity entered is greater than the set for the characters, the text will be pushed to the left, in order to ensure the maximum amount of characters set in the option of **Values formatting**.

Validation Image

When you enable this option, an image will be shown next to the field informing whether the field is in accordance with the settings of *minimum size* and *Maximum* size (as you can see in the images below) defined in the Formatting value option.

In the example below, the field was set to receive at least 5 characters, see what happens with the image when it receives only 4 characters:

However, if the value inserted has 5 or more characters the the image will change according to example bellow:

Use slider:

It displays a slider component in the field. So you can increase or decrease the value sliding the cursor. You can also customize the increment value, if it increments the value 1 by 1, 2, 5, 10... N.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Validate with Mask

By enabling this option you will be able to enter data according to the mask that was configured in the option **Field Mask** and Scriptcase will validate it.

Record Variable

Allows you to record a session variable with the value of the field ([global variable] [var_glob]), to be used in other applications.

Example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable Name

In this option you must set the session variable name, enabled in the previous item, which will receive the value of the field.

You need to inform only the variable name, for example: **var_seller**.

The recovery of the value is made as [Global Variable][var_glob]{:target='blank'}.

Field size in Database

This option sets the size of the field relative to the size that is configured in the database. By default this value is already configured automatically when the application is generated.

Hidden Field

This option when enabled will hide the field inside the application at the time of execution.

Label Field

By enabling this option, the field will be changed to only one label where the information will be displayed, so it is not possible to make changes or inserts in the field configured as label.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disable field

Allows you to disable the field, making it impossible for the user to enter a value according to the developer-defined option.

The available options are:

No - initial value of the attribute, this option does not disable the field. **Update mode** - This option disables the field only when editing the records. **Insert mode** - This option disables the field only inserting new records. **Update/Insert mode** - This option disables the field in both editing and inserting new records.

HTML Type

HTML object used to display the field in the application.

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:

Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

```
SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}
```

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

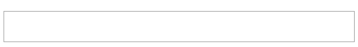
Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.



Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of

Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Decimal Field

General Settings

Decimal field configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser’s autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser’s autocomplete.
- When **enabled**, the field’s autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit grouping

It defines if the field displays the digits separator.

Maximum Size

It defines the max size of the field.

Minimum Value

It defines the min value of the field.

Maximum Value

It defines the max value of the field.

Decimal Precision

It defines the number of decimal places for the field.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Accept

It determines if the field accepts only negative, positive, or both numbers.

Show a calculator

It displays an icon beside the field to help the user calculating the field value.

Auto-complete with zeros

It automatically informs the decimal places when not specified. If disabled, the user always needs to inform the decimal value.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.

- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is

being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Currency Field

General Settings

Currency field configuration Interface.

Data Type

Define the field type to Currency. It allows defining the currency number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser’s autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser’s autocomplete.
- When **enabled**, the field’s autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit grouping

It defines if the field displays the digits separator.

Currency symbol usage

It defines if the field displays the Currency Symbol of the Regional Settings.

Maximum Size

It defines the max size of the field.

Minimum Value

It defines the min value of the field.

Maximum Value

It defines the max value of the field.

Decimal Precision

It defines the number of decimal places for the field.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Accept

It determines if the field accepts only negative, positive, or both numbers.

Show a calculator

It displays an icon beside the field to help the user calculating the field value.

Auto-complete with zeros

It automatically informs the decimal places when not specified. If disabled, the user always needs to inform the decimal value.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - HTML Image Field

General Settings

HTML Image field Configuration Interface.

Data Type

Define the type of field. When setting it to HTML Image, it allows to display an image into the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Image

Set it to display an image. The icon “Select Image” lists all images from Scriptcase and your uploaded images. The “Upload an image” option allows you to send a copy to the Scriptcase server.

Border

Define the width of the Image border in Pixels.

Width

Define the image width size in Pixels.

Height

Define the image height size in Pixels.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.

- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is

being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Label Field

General Settings

Label field Configuration Interface.

Data Type

Define the type of field. When setting it to label, it displays the value with no input to change it.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Text

You can inform a text to be displayed beside the field.

Reload

It links the informed text to reload the form when clicking on it.

Position

It allows positioning the field in the block.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.

- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Card Field

General Settings

Credit Card Number Configuration Interface.

- **Data Type** : Defines the type of field for the application. When it is defined as a Credit Card Number, the field verifies if the value is valid.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Creates a placeholder on the field with the text informed.
- **Initial Value (type)** : Set the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Defines the amount of characters allowed for the field.
- **Validation Image** : Display an image next to the field when it is being validated.
- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to determine the max amount of characters to be typed in.
- **Field Mask** : Defines the display mask for the field. There are three types of masks that can be merged.

Character	Description
9	Represents a numeric character (0-9)
A	Represents an alpha numeric character (A-Z,a-z)
*	Represents any alpha numeric character (A-Z,a-z,0-9) typed by the user.

It is possible to merge two or more masks simultaneously, separated by a semi coma with the smallest mask at start. The replacement occurs when the user is typing when the lowest amount of character exceeds.

Examples of Masks

Field	Mask	Informed Value	Formatted Value
Telephone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Software Key (Only Numbers)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters an Numbers)	**---**	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plates	AAA - 9999	QWE1234	QWE - 1234
ScriptCase License	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple Masks (Telephone)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

- **Hidden Field** : This option makes the field hidden, but still allows it is value to be processed through JavaScript or PHP.
- **Label Field** : This option make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Display the data type of field in the database.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.

- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Card Type Field

General Settings

Credit Card field configuration Interface.

Data Type

Define the field type to Credit Card. It sets a combo-box for selecting the credit card type.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Field size in database

It determines the field size in bytes. It limits the max quantity of the allowed characters.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

 Label Configuration Interface.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn't chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn't chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn't chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Lookups Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - E-mail Field

General Settings

Email field configuration Interface.

- **Data Type** : Defines the type of field for the application. When set to Email, the field applies validations for an email format.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Defines the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.

- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Defines the amount of characters allowed for the field.
- **Validation Image** : Displays an image next to the field when it is being validated.
- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Defines the name for the session variable that will receive the field value.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through Javascript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Display Icon** : Displays the Email icon next to the field.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.

- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - URL Field

General Settings

URL field configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Display Icon

Displays an icon beside the field.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

Show the URL as a clickable link

It sets the content of the field as a clickable link.

Target handling where the link will open

Target to open the link when clicking on it.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

```
SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
```

The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input

Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.

- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- n template
- form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - HTML Color Field

General Settings

HTML Color field configuration Interface.

- **Data Type** : Select the type of field for the application. When set to HTML Color, you can select a color to be used in the form.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Set the initial value to the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Defines the amount of characters allowed for the field.
- **Validation Image** : Displays an image next to the field when it is being validated.
- **Save Variable** : Save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Defines the name for the session variable that will receive the field value.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of the field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.

- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos ▷

Control - HTML Editor Field

General Settings

HTML Editor field configuration Interface.

- **Data Type** : Defines the type of field for the application. When set to HTML Editor, you can inform any type of character and they will be saved in HTML form.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Height** : Set the Height in pixels to the HTML Editor field.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through Javascript or PHP.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **SQL Type** : Displays the data type of field in the database.

Toolbar

HTML Editor toolbar settings Interface.

- **Properties**
  - **Position** : Location of the HTML Editor toolbar.
  - **Alignment** : Toolbar button alignment.
  - **Status** : Status Bar Display (Do not Display, Top and Bottom).
  - **Amount** : The Amount of lines of the HTML Editor toolbar.
- **Button Organization** : Positions the toolbar buttons.
- **Preview** : You can visualize the toolbar according to the previous settings.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**

- **Font** : Allows to choose the font type, that will be applied to the application field title.
- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Localization Field

General Settings

Location field Configuration Interface.

Data Type

Define the type of field. When setting it to Location, you can see a Combobox with the Languages that are part of the Project.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Reload

It links the informed text to reload the form when clicking on it.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Theme Field

General Settings

Theme field configuration Interface.

Data Type

Define the field type to Theme. it shows a combobox with the list of themes set in your project.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Reload

It Reloads the form when modifying the value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**

- **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- n template
- form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Date Field

General Settings

Date field configuration Interface.

Data Type

Define the type of field. When setting it to Date, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server Date.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in "Insert Mode", "Update Mode" or in "Insert and Update Mode".

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Date separator

It allows you to inform the separator symbol for the date.

First Day

Defines the first day of the week.

Display

It offers predefined formats for displaying dates.

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

Use Combo-box

It allows you to select the date from a combo-box.

Minimum Date

- **Fixed Date**: Inform the minimum date allowed presented in the selected date format.
- **Current Date**: By clicking on the icon next to the text field, ScriptCase offers the following options:
  1. **Actual date**: Set the current date as the minimum allowed date.
  2. **Actual date with increment**: Set the minimum date with the current date more(+) the days, months, or years you want to increment.
  3. **Actual date with decrement**: Set the minimum date with the current date minus(-) the days, months, or years you want to decrement.

Maximum Date

- **Fixed Date**: Inform the maximum date allowed presented in the selected date format.
- **Current Date**: By clicking on the icon next to the text field, ScriptCase offers the following options:
  1. **Actual date**: Set the current date as the maximum allowed date.
  2. **Actual date with increment**: Set the maximum date with the current date more(+) the days, months, or years you want to increment.

3. **Actual date with decrement**: Set the maximum date with the current date minus(-) the days, months, or years you want to decrement.

Display Format

It enables the Date format beside the field when informing the date.

Date display position

Available when activating the **Display Format** flag, defines where the filling mode will be displayed. The options are: **Right**, **B Lower** and **Watermark**.

Display Calendar

It enables a calendar icon beside the field that allows selecting the date from a calendar to fill the input.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.

    - **Height** : To define a height for the title of the field.
- **CSS of the Field**
    - **Font** : Allows to choose the font type, that will be applied to the application field.
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the field.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
    - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the field.
    - **Height** : To define a height for the field.
- **CSS of the Input Object**
    - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
    - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
    - **Font Color** :Allows to choose a color for the font from the color pallet.
    - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
    - **Bold** : Applies the bold style to the font.
    - **Underline** : Applies the underline style to the font.
    - **Border Size** : Applies the border size to the Input Object.
    - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
    - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
    - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
    - **Width** : To define a width for the Input Object.
    - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be

displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Time Field

General Settings

Time field configuration Interface.

Data Type

Define the type of field. When setting it to Time, you can inform a time format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in "Insert Mode", "Update Mode" or in "Insert and Update Mode".

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the Time. When not enabled, it displays the Time separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying Time.

Internal Format

It allows defining the field format when the SQL type is different from **TIME**. You must use the characters **HH**, **II**, and **SS** that correspond to **Day**, **Hour**, **Minutes** and **Seconds**.

Minimum time

Minimum value accepted by the field.

- **Fixed Time** : Enter the minimum time that scriptcase will criticize in the time type field in the format as presented.
    - **Current Time** : When clicking on the icon next to the box, scriptcase provides the following options:
    - **Current Time with Increment** : The minimum time will be the current time (+) the days or months or years that you increment.
    - **Current Time with Decrement** : The minimum time will be the current time (-) the days or months or years that you want to decrement.

Maximum time

Maximum value accepted by the field.

- **Fixed Time**: Manually enter the maximum time that scriptcase will input in the time type field in the format as it is presented.
    - **Current Time**: When you click the icon next to the box, scriptcase provides the following options:
    - **Current Time with Increment**: The maximum time will be the current time (+) the days or months or years that you increment.
    - **Current Time with Decrement**: The maximum time will be the current time (-) the days or months or years that you want to decrement.

Display Format

It enables the Time format beside the field when informing the time.

Display position

Available when activating the **Display Format** flag, defines where the filling mode will be displayed. The options are: **Right**, **B Lower** and **Watermark**.

Display full name

Displays the full name instead of the mask. Ex: day month year.

Use Time picker

It uses the JQuery plug-in to choose the time.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Date and time Field

General Settings

Date and Time field configuration Interface.

Data Type

Define the type of field. When setting it to Date and Time, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server Date and Time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser’s autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser’s autocomplete.
- When **enabled**, the field’s autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Regional settings

Allows you to apply regional date formatting settings to the field. When not selected, the Date Separator and Time Separator attributes will be displayed.

Display

Allows you to select the display format of the date/time field.

Internal Format

Field storage format in the database.

Minimum Date

Minimum value accepted as input for the field.

- **Minimum Date** :
  - **Fixed Date** : Enter the minimum date that scriptcase will criticize in the date type field in the format as presented.
  - **Current Date** : When clicking on the icon next to the box, scriptcase provides the following options:
  - **Simple Current Date** : Will leave the current date as the maximum date, that is, no one born after the current day will enter the form.
  - **Current Date with Increment** : The minimum date will be the current date (+) the days or months or years that you want to increment.
  - **Current Date with Decrement** : The minimum date will be the current date (-) the days or months or years that you want to decrement.

Maximum Date

Maximum value accepted as input for the field.

- **Maximum Date** :
  - **Fixed Date** : Enter the maximum date manually that scriptcase will criticize in the date type field in the format as it is presented.
  - **Current Date** : When clicking the icon next to the box, scriptcase provides the following options:
  - **Simple Current Date** : Will leave the current date as the maximum date, that is, no one born after the current day will enter the form.
  - **Current Date with Increment** : The maximum date will be the current date (+) the days or months or years that you want to increment.
  - **Current Date with Decrement** : The maximum date will be the current date (-) the days or months or years that you want to decrement.
  - **Display Format** : Enables display of the mode

Minimum time

Maximum value accepted by the field.

- **Fixed Time**: Manually enter the maximum time that scriptcase will input in the time type field in the format as it is presented.
  - **Current Time**: When you click the icon next to the box, scriptcase provides the following options:
  - **Current Time with Increment**: The maximum time will be the current time (+) the days or months or years that you increment.
  - **Current Time with Decrement**: The maximum time will be the current time (-) the days or months or years that you want to decrement.

Maximum time

Maximum value accepted by the field.

- **Fixed Time**: Manually enter the maximum time that scriptcase will input in the time type field in the format as it is presented.
  - **Current Time**: When you click the icon next to the box, scriptcase provides the following options:
  - **Current Time with Increment**: The maximum time will be the current time (+) the days or months or years that you increment.
  - **Current Time with Decrement**: The maximum time will be the current time (-) the days or months or years that you want to decrement.

Hour value

Sets the value to be assigned to the time when it is empty and a date is selected in the date picker.

- **Available values are**: Start of day, End of day and Current time.

Apply date limits to the calendar

Apply Minimum and Maximum Date limits when selecting calendar dates, avoiding the selection of dates outside these limits.

Display Format

Display the field format in the data cell.

Date display position

Available when activating the **Display Format** flag, defines where the fill mode will be displayed. The options are: **Right**, **Below** and **Watermark**.

Display full name

Displays the full name instead of the mask. Ex: day month year.

Group Date and Time

Group date and time in the same field.

Display Calendar

Allows you to display a calendar icon next to the field. This allows you to select the month and year, passing the value defined in the calendar to the field.

New Calendar

Show a new calendar with jquery, or the old calendar.

Years Limit

Number of years that will be displayed on the calendar.

View week number

Display week number in application.

Additional months

Display additional months in the calendar.

Available options are: 1, 2 and 3.

Show Combo year and month

Show Combo Year and month in calendar

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.

- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Image (File Name) Field

General Settings

Data Type

Define the type of field. When setting it to Image (File Name), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Image Border

The width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Progress bar

It displays a progress bar when sending the files to the server.

Upload area

It displays a drag and drop area to upload the file.

Upload area icon

Font-awesome icon for viewing in the file upload zone.

Clickable upload area

Hides the upload button and turns the upload area responsible for the upload function.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API

Keep file after uploading via API

It also saves the file in the default folder for images or documents after uploading to a cloud storage API.

Deletion in the storage API

When deleted in the application, the file will also be removed from the folder in the cloud storage API.

Increment file

It inserts a number increment into the file name if there is already another one with the same name in the upload folder.

Subdirectory for local storage

It sets the Sub-folder name of the stored files. It is relative to the directory of Document upload (see the Settings). It is possible to use global variables or local variables to format the name of the sub-folder.

Create Subfolder

It creates the sub-folder if not already created.

Image Caching

Time in minutes to store the image in the cache.

Hide image name

It displays only the image, without the name.

Files Deletion

It Deletes the files from the directory when deleting a record from the database.

Display link only

It allows showing a link to open the image in another window with the original size. (with no re-dimensioning)

Open in Another Window

Allows to open the image in another window.

File Size

It allows defining the field to store the file size in the database.

Extensions and upload size

This setting defines the extensions allowed for uploading and the maximum size of each one.

This setting is important for the security of your project, as it prevents unwanted files from being uploaded.

> **Note** If no value is informed, all extensions will be allowed. The file size will be limited by your PHP configuration.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- n template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Document (File Name) Field

General Settings

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

 Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.
- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]
- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.
- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction

Related Videos

- Control overview
- Form overview

Control - Select Field

General Settings

Select field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Select, you can select multiple option from a combo box (Select Field).
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.
- **Use Select2** : Uses the new component for data selection, allowing searches within the select.
- **Display Select2 search area** : Sets whether to display the search field within Select2.

Initial Value Configuration Interface.

- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the field size in bytes. It is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through Javascript or PHP.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

  Selecting the lookup type.

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

The value of the key_field will be stored in the table field.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimitation.

  - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Height** : Defines the height for the select object.
- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.
- **Link** : Allows to create a link to another form allowing to manipulate the list displayed on the select field. After the manipulation, the select object it updated automatically.

◦ **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Height** : Defines the height for the select object.
  - **Multiple Values (delimiter)**

    You can store various values for the select field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Height** : Defines the height for the select object.

- **Multiple Values (position)**

  Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

  For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

  The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

  Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

  Setting up Multiple Values (position)

  - **Label** : Text that will be displayed in the item list of the select field.
  - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
  - **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
  - **Start** : Starting position of the string that is going to be stored. The first position is always 1.
  - **Size** : Amount of bytes that is going to occupy in the string.
  - **Height** : Defines the height for the select object.

- **Multiple Values (binary)**

  Allows to store a diversity of information selected on the field in only one value.

  In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  To store the data, it is done an internal sum of the decimal number. See the examples below.

  Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports

Attribute Value	Lookup Description
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

Editing Lookup Configuration Multiple Values (Binary)

- **Label** : Text that will be displayed in the item list of the select field.
- **Height** : Defines the height for the select object.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

Displaying the original lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Lookup Method - Actual value

This lookup is used to list all the values in the selected field.

This lookup will apply a “distinct” to your SQL query.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template

Related Videos

- Responsive login template
- Control overview
- Form overview

Control - Double Select Field

General Settings

Double Select field configuration Interface.

- **Data Type** : Select the type of field for the application. When set to Double Select you can have multiple options selected.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Defines the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to set the max amount of characters to be typed in.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through Javascript or PHP.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Displays the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, getting these values from the database.

Lookup Settings Display for the field.
Automatic Lookup Interface..

- **SQL Select Statement** : Defines the SQL command that will get the values displayed on the form field. The command needs to have the following structure:

  ```
  SELECT key_field , display_field FROM table
  ```

  The value of the key_field will be stored in the table field.
- **Height** : Set the height(lines) of the field interface.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter than (;).
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the

users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

Related Videos ▷

- nguage selection
- Contact Form template
- Registration form template
- Control introduction
- Responsive login template
- Control overview
- Form overview

Control - CheckBox Field

General Settings

CheckBox field configuration Interface.

- **Data Type** : Select the type of field for the application. When set to CheckBox, you can have multiple options selected.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Defines the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to determine the max amount of characters to be typed in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through JavaScript or PHP. .
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query). *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the CheckBox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Reload form when value has changed** : With this option selected, the form will refresh if

there is a change to the selected object in the field.

- **Option check all** : Displays two options on the field to check and uncheck all.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).

  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Columns** : Set the amount of columns, for the list of items.

  - **Multiple Values (delimiter)**

    You can store various values for the checkBox field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Columns** : Set amount of columns, for the list of items.

  - **Multiple Values (position)**

    Stores a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

    For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

    The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1

Label	Value	Start	Size
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Columns** : Set the amount of columns, for the list of items.

- **Multiple Values (binary)**

Stores a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

Setting up Multiple Values (Binary)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Columns** : Allows you to inform the amount of columns, for the list of items.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Saves all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Refreshes the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Option check all** : Displays two options on the field to check and uncheck all.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.

- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Radio Field

General Settings

Radio field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Radio, your allowed to select one of the options listed.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to determine the max amount of characters to be typed in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through Javascript or PHP. .
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query). *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.
    - **Columns** : Allows you to inform the amount of columns, for the list of items.

- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

◦ **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Label** : Text that will be displayed in the item list of the radio.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Default** : With this option enabled, the selected item will be inserted to the table when committed.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on load lookup definition.
- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

Displaying the original lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Settings

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection

Related Videos ▷

- n template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Control - Profile Field

General Settings

Profile field configuration Interface.

Data Type

Define the type of field. When setting it to Profile, it displays a Combobox with the profiles set up in the project properties.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.

- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed

- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- n template
- form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Menu Links Field

General Settings

Menu Links field Configuration Interface.

Data Type

Define the type of field. When setting it to Menu Links, you can show a list of links in the application.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

 Values Format Interface.

It controls the Insert, Update and Delete of the menu items.

Label

The Label option lets you define the title of a field.

Link

It allows defining to call an external link when clicking on the item.

Icon

It displays an icon beside the link.

Default

Set the current item as default.

Option Buttons

- **Insert**: Insert a new item into the menu.
- **Update**: Update the selected item.
- **Delete**: Delete the selected item.
- **Clear**: Clears the data of the selected item.
- **Clear All**: Removes all items from the menu.

Display Mode

Define how to display the menu items; Select, Radio or Link:

- **Select**:

- **Radio**:

- **Link**:

Hide Label

You can enable or disable the label displaying.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.

- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse

leaves the help icon.

- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Control Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Text Auto-Complete Field

General Settings

Text auto complete field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Text auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal Text for the data.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You’ll inform the Defined Value here.
- **Use Select2**: Uses the new component for data selection, allowing searches within the select.
- **Amount of characters for the Select2** : Sets the number of characters to start the search in Select2.
- **Amount of lines for the Select2** : Sets the maximum number of rows to list the search result in Select2.
- **Width for the Select2** : Sets a width for the area for the Select2.

Initial Value Configuration Interface.

- **Amount of characters** : Define the amount of characters allowed for the field.
- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Complete to the Left** : Allows to define the character that will be used to complete the value to the left that the user typed in to the max size of the field defined in the Field size in database option.
- **Field size in database** : Determines the field size in bytes. it is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Validate on submit** : Validate the field only when the form is submitted.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

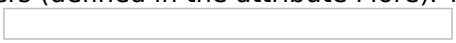
Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed. 

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

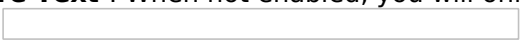
Error message

Message to be displayed if the regular expression is not valid.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).
- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below.  *lookup Settings configuration Interface.*
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  lookup Settings validation configuration Interface.
- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links 🔗

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction

Related Videos ▷

- Control overview
- Form overview

Control - Number Auto-Complete Field

General Settings

Number auto complete field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Number auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal number for the data.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Define the amount of characters allowed for the field.
- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Complete to the Left** : Allows to define the character that will be used to complete the value to the left that the user typed in to the max size of the field defined in the Field size in database option.
- **Field size in database** : Determines the field size in bytes. It is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Validate on submit** : Validate the field only when the form is submitted.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed.

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).
- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below. *lookup Settings configuration Interface.*
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  lookup Settings validation configuration Interface.
- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Signature Field

General Settings

Configuration Interface of the Signature Field.

The signature field will help you creating more sofisticated forms and making it possible to store signatures in your database. Inside our development environment we have specific settings that will help you to customize your field, those options are:

- **Data Type** : You can define the type of field for the application. When it is defined as a text, it accepts letters, numbers and special characters.
- **Label** : Lets you define a label to the field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Background color** : Defines a color to the field background by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Pen color**: Set a color to the pen by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Width** : Lets you define a width to the field.
- **Height** : Set a height to the field.
- **Subtitle** : Defines the subtitle that will be displayed beside the field.
- **Initial Value** : Lets you define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
- **Save Variable** : Save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Disabled Field** : Define if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : Displays the HTML Object that will be used to display the field in the form.
- **SQL Type** : Displays the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of ScriptCase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **CSS of the Title**
  - **Font** : Select the font type, that will be applied to the application field title.
  - **Font Size** : Defines the font size, that will be applied to the application field title.
  - **Font Color** : Choose a color to the font by using a valid hexadecimal color value or from the color picker.
  - **Background Color** : You can define the color for the field by using a valid hexadecimal color value or from the color picker.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Set the underline style to the font.
  - **Border style** : Choose a style for the border.
  - **Collapse** : Defines the collapse for the border.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Choose a color for the border, using a color palette to apply to the title.
  - **Horizontal Alignment** : Position the label of the field in the wanted position (left,right,center and justify).
  - **Vertical Alignment** : Position the label of the field in the wanted position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : Defines a width for the title of the field.
  - **Height** : Set a height for the title of the field.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

Related Videos ▷

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Control - Rating Field

General Settings

This field allows the developer to create a field for rating using stars (or any other image), where the final user can select a rate option.

Attribute descriptions

Data Type

Define the field type to **Rating**.

Label

The Label option lets you define the title of a field. Example: If the database field name is “**Stars**”, You can display a different name for the user, like “**Stars**”.

Besides use a fixed text, the **Label** attribute allows the use of **Langs** to define the field title, allowing the internationalization of your application.

Label below field

Defines the message to be displayed below the field.

Subtitle

Define the subtitle of the field, below the ratings. **Example**: “Thank you for your feedback!”.

As in the **Label**, the **subtitle** attribute also allows the use of **Langs** for the internationalization of your application.

Amount of icons

It defines how many icons it will display in the field. The value set in this attribute must be according to the evaluation rules.

Initial Value (type)

Allow the definition of an initial value for the field when the form application is in insert mode. The only option available in this field is **Defined**. Selecting this option the attribute **Initial Value** will be showed for set the value.

Example: The initial value is 3 when inserting a new record, the rate field will initiate with 3 stars already selected.

> The value defined in this attribute will overlap any value defined previously.

Save variable

It allows saving the value of the field in a session variable(Global variable) to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the **session variable** that will receive the field value.

We must inform only the variable name, - **var_rating**.

We can get the value as a global variable.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of ScriptCase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **CSS of the Title**
  - **Font** : Select the font type, that will be applied to the application field title.
  - **Font Size** : Defines the font size, that will be applied to the application field title.
  - **Font Color** : Choose a color to the font by using a valid hexadecimal color value or from the color picker.
  - **Background Color** : You can define the color for the field by using a valid hexadecimal color value or from the color picker.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Set the underline style to the font.
  - **Border style** : Choose a style for the border.
  - **Collapse** : Defines the collapse for the border.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Choose a color for the border, using a color palette to apply to the title.
  - **Horizontal Alignment** : Position the label of the field in the wanted position (left,right,center and justify).
  - **Vertical Alignment** : Position the label of the field in the wanted position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : Defines a width for the title of the field.
  - **Height** : Set a height for the title of the field.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control - Rating Smile Field

General settings

This field type allows the final user evaluate with an emoji that goes from **Very Bad** to **Excellent**.

Data type

Defines the data type of the field in the application. In this case we should select **Rating Smile**.

Label

Defines the title that will be shown in the field when the application is running. The used terms in the interface are essential to the system has good usability, we should use comum name and terms for the final user, instead of the default values of the system.

For example, for the rating smile field, the client would have a better understanding of the field function when the label is **Product evaluation**.

Besides a fixed text, the **Label** allow to use of lang to define the field title, this way is possible to use the application internationalization.

Label below field

Allows you to define a message that will be displayed under the field, which can serve as a help or caption for example.

Subtitle

Define a subtitle that will be shown beside the field. Ex.: Thank you for the feedback!

Below is one example of an application subtitle:

Values and hints

Allow to the developer to define the values to each emoji taht will be saved at the data base and allow to define a hint **(It is show passing the mouse above the emoji)**

Display format

Allows the developer to define the format that the emojis will be shown.

In case the first option is defined, the emojis will be shown like this:

In case the second option is defined, the emojis will be shown like this:

Icon size

Allows defining the icon size in pixels.

Rating padding

Allows for defining the padding value between the emojis.

Icon colors

Allows the icon colors to be defined in each emoji.

Initial value (type)

Allows defining a default value to the field when the application will be in insert mode. Is possible to choose between two options:

Default value: Selecting this option, the default value will be available, where we should input the field's default value. For example, my default value is **Allan**, inserting a new register, the field **Seller Name** will start with Allan.

Save variable

It allows saving a session variable (global variable) with the field value, to be used in other applications.

For example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable name

In this one we should define the variable's name in session, available at the last item, which will receive the field value.

We should inform only the variable's name, **var_seller**.

The recovery of the value is done in the form of a global variable.

Disable field

Allow to Disable field, impossible to change the defined value.

Available options are:
No - Default value, this option don't disable the field.
Update mode - This option disable the field only at update. **Insert mode** -This option disable the field only at insert. **Insert/Update mode** -This option disable the field in both insert and update.

HTML type

HTML object used to show the form field.

SQL type

Infomrs the field type at the data base.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Control - Rating Thumbs Field

General settings

This field type allows the final user to save data with the icons like or dislike.

Data type

Defines the data type of the field in the application. In this case we should select **Rating Thumbs**.

Label

Defines the title that will be shown in the field when the application is running. The used terms in the interface are essential to the system has good usability, we should use comum name and terms for the final user, instead of the default values of the system.

For example, for the rating thumbs field, the client would have a better understanding of the field function when the label is **Product evaluation**.

Besides a fixed text, the **Label** allow to use of lang to define the field title, this way is possible to use the application internationalization.

Label below field

Allows you to define a message that will be displayed under the field, which can serve as a help or caption for example.

Subtitle

Define a subtitle that will be shown beside the field. Ex.: Welcome to help us with your feedback!

Below is one example of an application subtitle:

Values and hints

Allow to the developer to define the values to each icon taht will be saved at the database and allow to define a hint **(It is shown passing the mouse above the icon)**

Display format

Allows the developer to define the format that the emojis will be shown.

In case the first option is defined, the emojis will be shown like this:

In case the second option is defined, the emojis will be shown like this:

Icon size

Allows defining the icon size in pixels.

Rating padding

Allows for defining the padding value between the emojis.

Icon colors

Allows the icon colors to be defined in each emoji.

Initial value (type)

Allows defining a default value to the field when the application will be in insert mode. Is possible to choose between two options:

Default value: Selecting this option, the default value will be available, where we should input the field's default value. For example, my default value is **Allan**, inserting a new register, the field **Seller Name** will start with Allan.

Save variable

It allows saving a session variable (global variable) with the field value, to be used in other applications.

For example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable name

In this one we should define the variable's name in session, available at the last item, which will receive the field value.

We should inform only the variable's name, **var_seller**.

The recovery of the value is done in the form of a global variable.

Disable field

Allow to Disable field, impossible to change the defined value.

Available options are:
No - Default value, this option don't disable the field.
Update mode - This option disable the field only at update. **Insert mode** -This option disable the field only at insert. **Insert/Update mode** -This option disable the field in both insert and update.

HTML type

HTML object used to show the form field.

SQL type

Infomrs the field type at the data base.

Display Sttings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Control Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Blocks

Blocks are “containers” where you can position the application fieldSlides of Forms, Controls, or Grids.

Scriptcase creates applications with one block by default. You can add more blocks as you wish, to organize it in the best way.

See below, the Columns Organization, and where you can define the position of the next block: beside or below the current one.

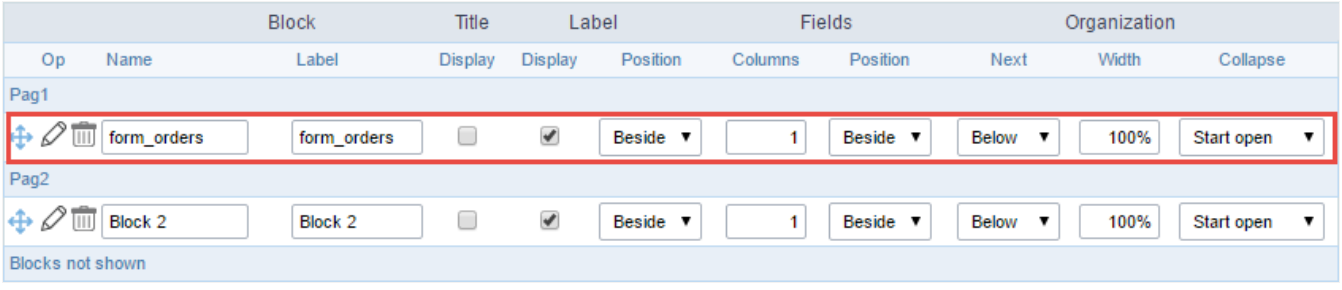
Application Block configuration

On the left side of each block, there are two icons, the first one to edit the information of the block and the second one to delete the block.

Organizing the position of the Blocks

See below how to modify the display order of the Blocks in one Page.

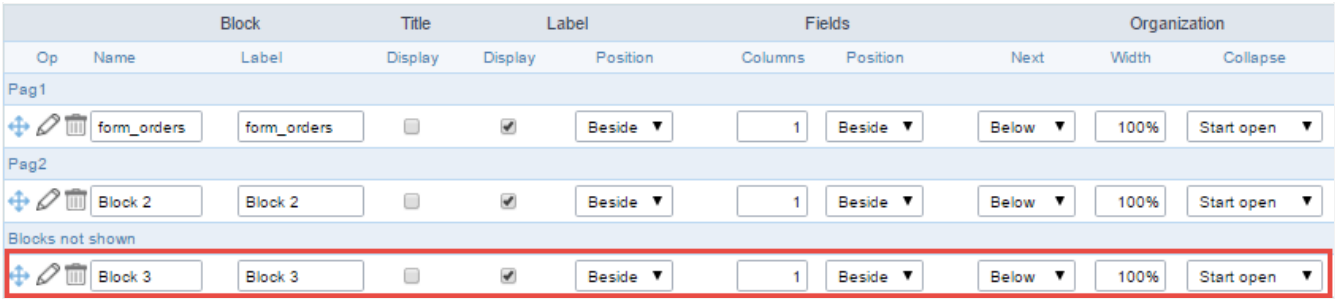
Click and drag the block that you desire to modify to its new position.



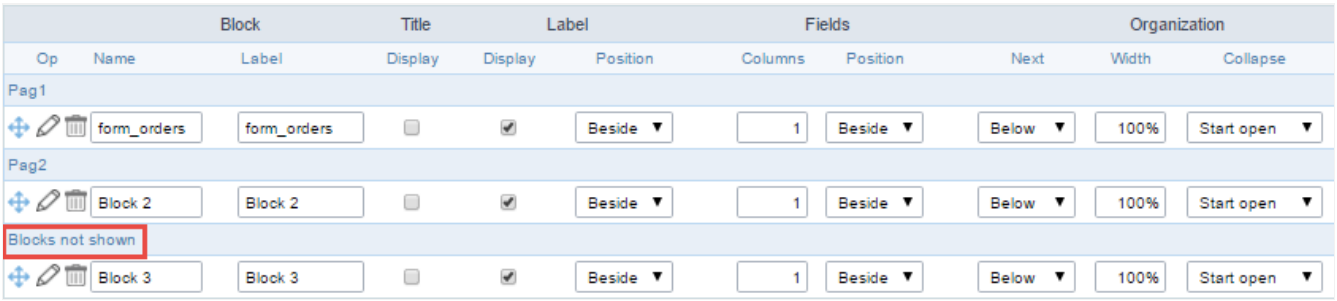
Application Block Display configuration

- See how to remove a block from the display

Click on the block desired and drag it to the item “Blocks not Shown”. This way, you can also drag the block to another page if desired. See the images below.



Application Block Display configuration



Application Block Display configuration

Block

- **Name**: The name that identifier the Block.
- **Label**: Title of the block to display in the application.

Title

- **Display**: It controls the display of the block title.

Label

- **Display**: It controls the display of the field labels of the block.
- **Position**: Options to display label :
    1. **Above**: Display the label above the field.
    2. **Beside**: Display the label beside the field.
    3. **Below**: Display the label below the field.

Fields

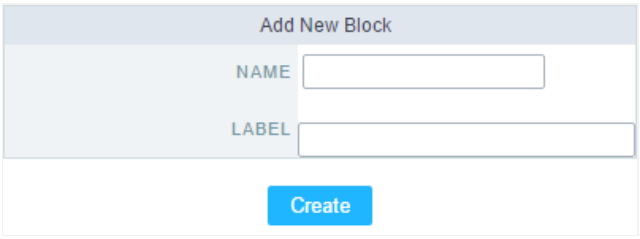
- **Columns**: Amount of columns side by side in the block.
- **Position**: The way to display the fields in the block:
    1. **Below**: Display the fields one below the other respecting the number of columns.
    2. **Beside**: Display the fields one beside the other respecting the number of columns.
    3. **Line**: Display the fields one beside the other with no tabulation.

Organization

- **Next**: The way to display the blocks in the page:
    1. **Below**: Set to show the following block below the current one.
    2. **Beside**: Set to show the following block beside the current one.
    3. **Tabs**: Set to show the following block in a different tab then the current one.
- **Width**: Set the block width in pixels or percentages. Use the symbol “%” to indicates the value in percentage.
- **Collapse**: Enables the option to close the block.

Create a New Block

To include new blocks in an Application, click on the button Create New Block. Then, enter the name and label of the block in the following interface and finish by click on Create.



Creating application blocks configuration

Name

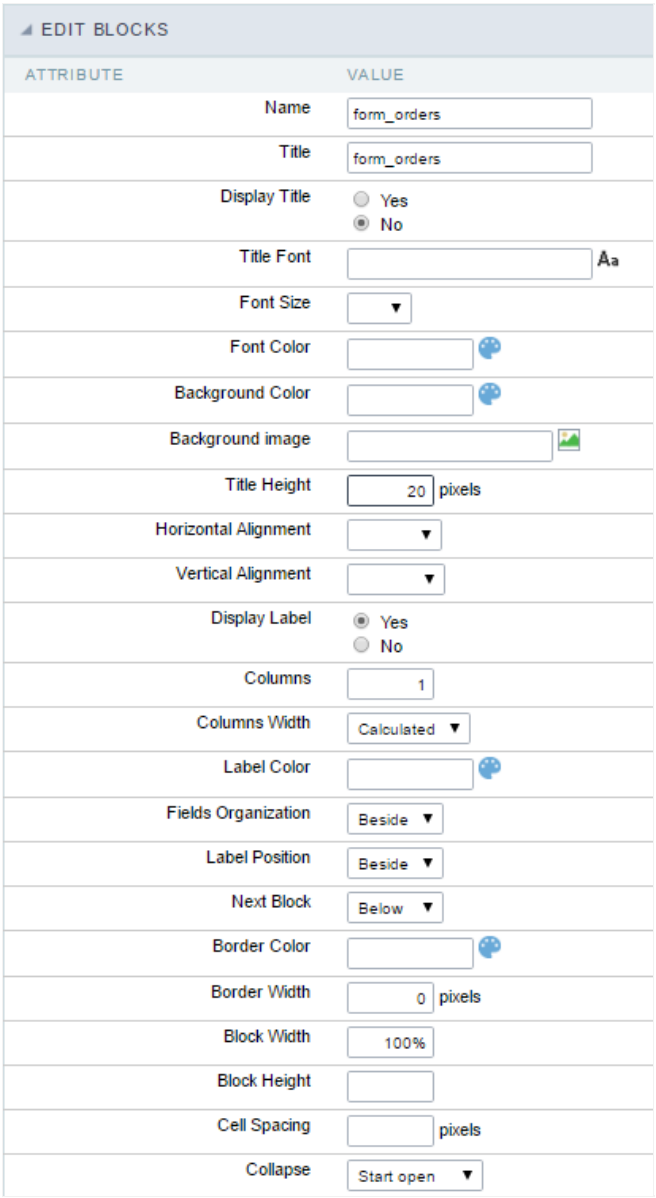
Name of the Block.

Label

Title of the block to display in the application.

Edit Blocks

To edit a block, click on the icon , that is on the left side of the block. Then you can see the following interface to define the parameters of the blocks. Click on Save to finish.



Application Block editing interface

Name

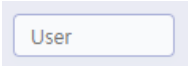
Name of the block. #### Title
Block title for display. #### Display Title
This option, when active, allows displaying the block title. #### Title Font
Set the font family of the block title. #### Font Size
Set the font size of the block title. #### Font Color
Set the font color of the block title. #### Background Color
Set the Background Color of the block title. #### Background image
Set a Background image for the block title. #### Title Height
Height in pixels of the block title line. #### Horizontal Alignment
Horizontal Alignment of the block title (Left, Center, and Right). #### Vertical Alignment
Vertical Alignment of the block title (Top, Middle, and Bottom). #### Display Label
Display the labels of the fields in the block. #### Columns
Amount of field columns in a block. #### Columns Width
Set the field column width of the block. #### Label Color
Color of the field labels. #### Fields Organization
The way to display the fields in the block. #### Label Position
Set the position of the field labels of the block.

The options are:

- **Beside** - This option positions the label on the right side of the field.

- **Above** - This option places the label above the field.

- **Below** - This option places the label below the field.

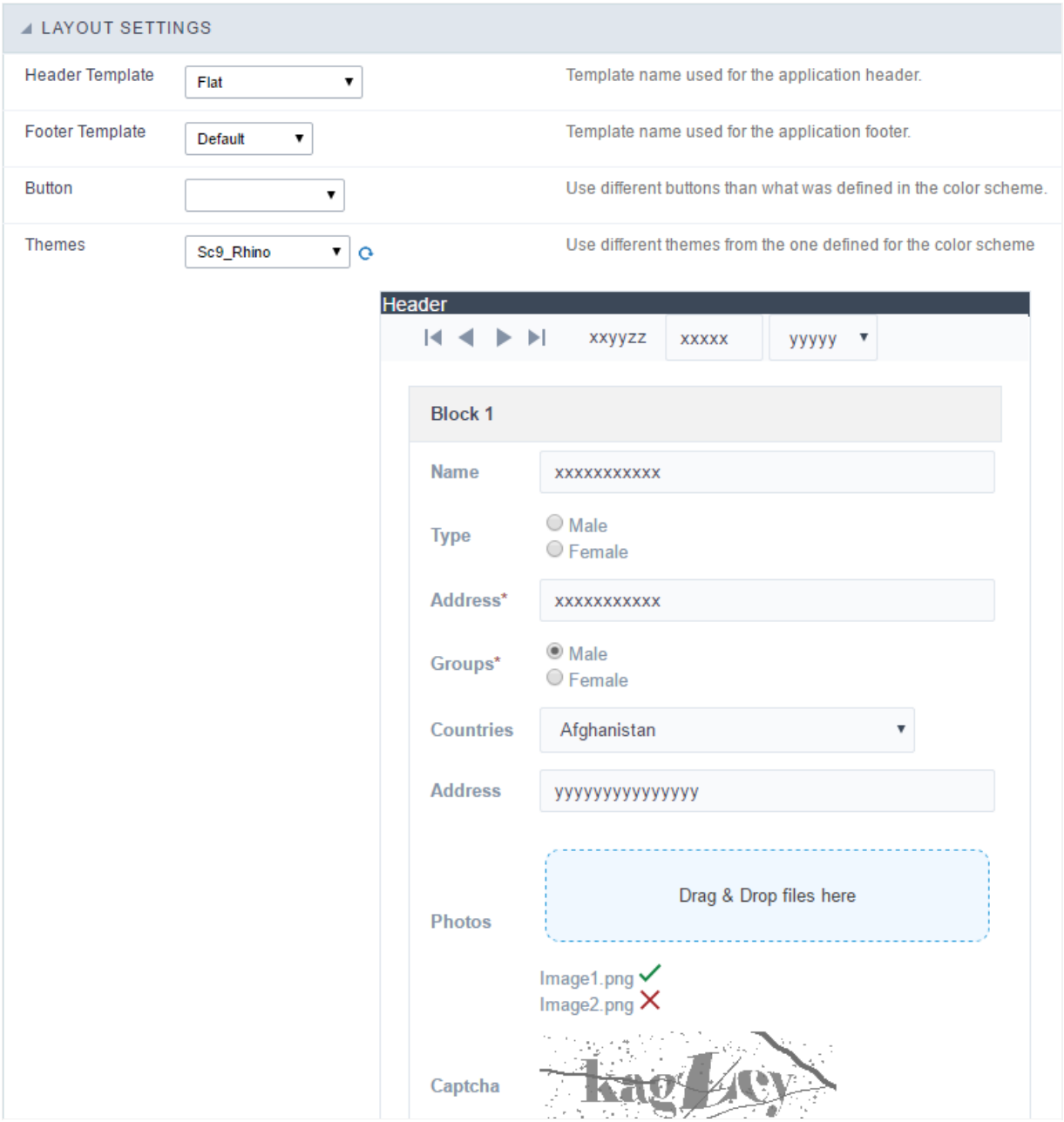
- **Watermark** - This option positions the label as a watermark. 

Next Block

Set the position of the following block relating to the current one. #### Border Color
The border Color for the block. #### Border Width
The border Width for the block. #### Block Width
The width for the block. #### Block Height
The Height for the block. #### Cell Spacing
The Cell Spacing in the block. #### Collapse
It enables the option to close the block.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout

Settings

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

Allows choosing the button theme for the application.

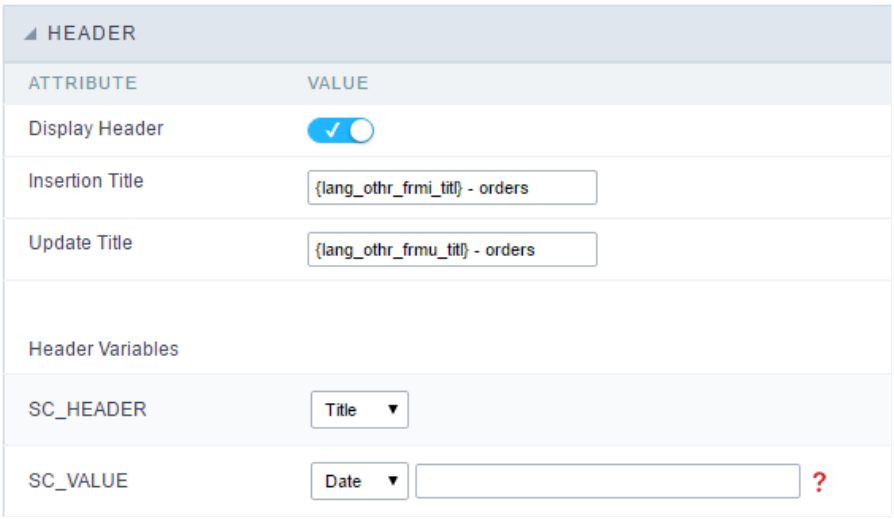
Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks

according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

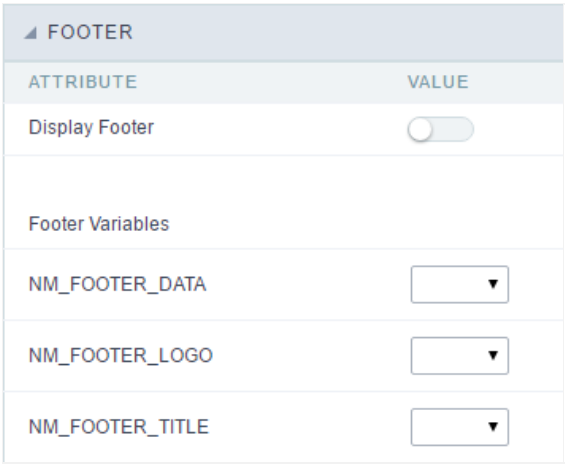
Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Related Links

Related Videos

Control Events Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application's CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event's PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Control

This event occurs before the application execute the SQL, and execute only once. Is used to do verification of variables, and security verification. Ex: if ([glo_var_department] != 'financial'){ sc_redir(app_x.php); }

onScriptInit - Control

These formulas are executed before the application starts. At this moment, the application locals variables are not available.

onLoad - Control

This event its executed before the form is loaded. In this moment all the applications variables are available.

onRefresh - Control

This event occurs when the form application is reloaded, is possible to reload the form based on fields (select , radio, checkbox)

onValidate - Control

This event occurs before the end user (inserts, updates or deletes) a row.

onValidateFailure - Control

This event occurs when submitting the application in OnValidate, there are errors generated by macros sc_error_exit, sc_error_message, or error detected by Scriptcase (errors related to Database).

onValidateSuccess - Control

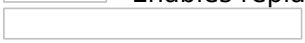
This event occurs when submitting the application to onValidate, there is no error generated by the macros sc_error_exit, sc_error_message, or by error verified by Scriptcase (errors related to Database).

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

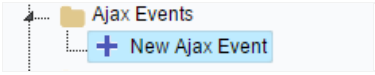
> Click Here to view the Scriptcase hotkeys documentation.

Control Ajax Events

OnClick

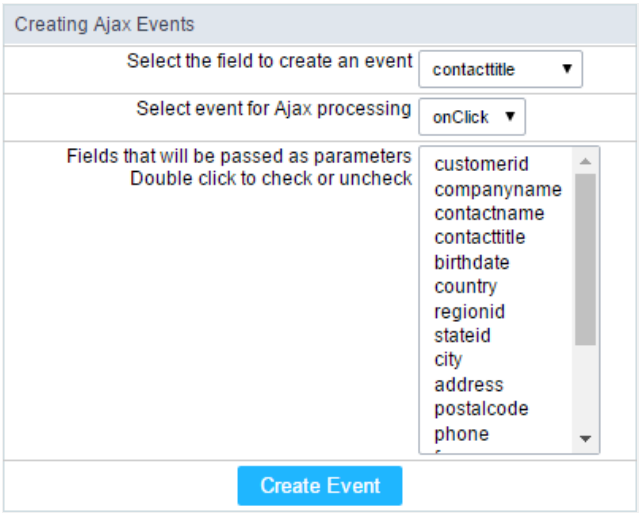
The ajax event OnClick is executed when the field that it's based on is clicked.

- Creating a new ajax event



Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

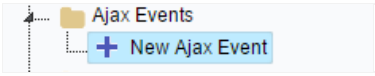
Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

OnChange

The ajax event OnChange is executed when the value of the field that it's based on is modified.

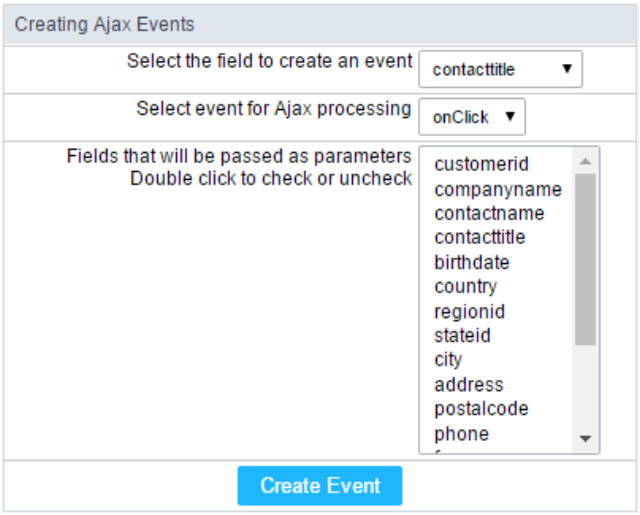
NOTE: The **radio field** is not compatible with this type of event.

- Creating a new ajax event



Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

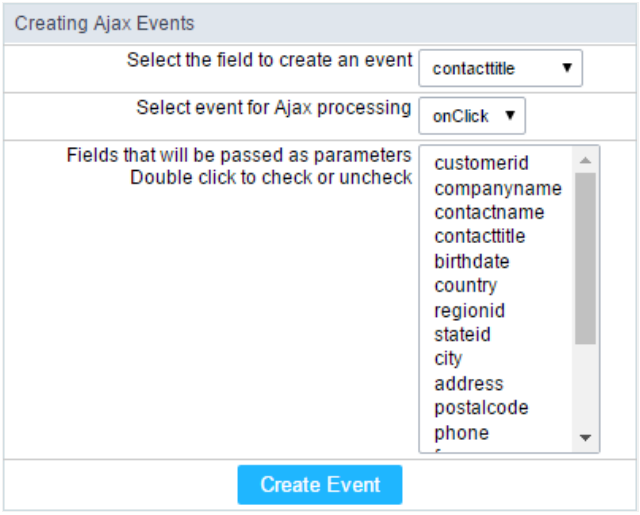
OnBlur

The ajax event OnBlur is executed when the focus is removed from the field that event is based on.

- Creating a new ajax event

Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

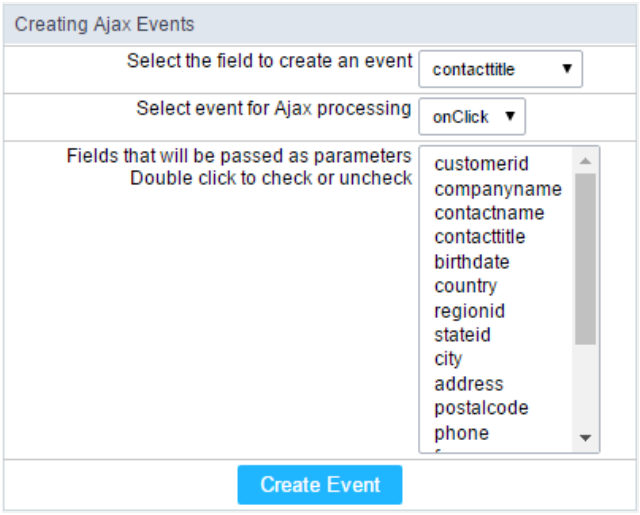
OnFocus

The ajax event OnFocus is executed when the field that it's based on is applied with a focus.

- Creating a new ajax event

Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.

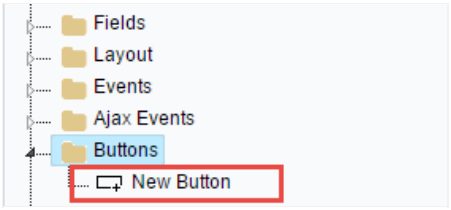
Choose an event that run the ajax Defines which event will be added to the field.

Related Links

Related Videos

Control Buttons Settings

In addition to the applications buttons, you can also create new manual buttons. These buttons will be placed on the application toolbar.

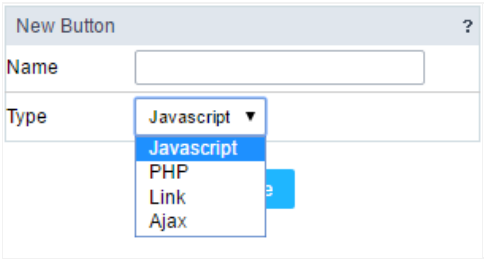


Creating new buttons

Creating a button

To create a new button, click the “new button” and type a name and the button type.

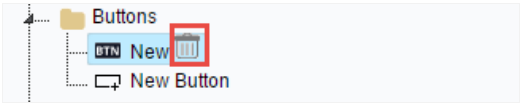
Form button types: (JavaScript, PHP, and Ajax).



Form button types

Deleting a button

To delete a button click on the icon next to the name of the button in the application menu (recycle bin).



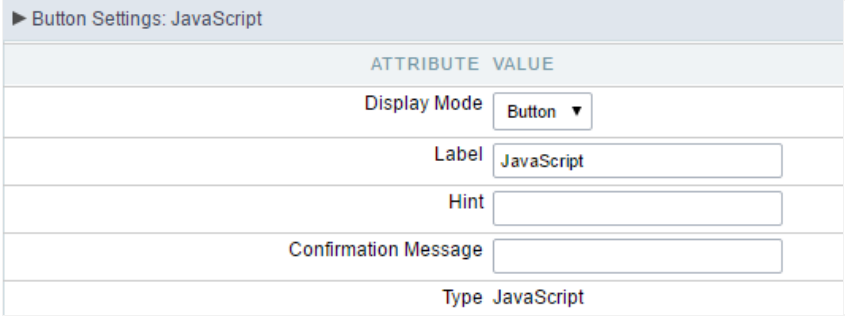
Deleting a button

JavaScript

Display Mode

You can configure the display mode of the javascript button in Image, Button or Link.

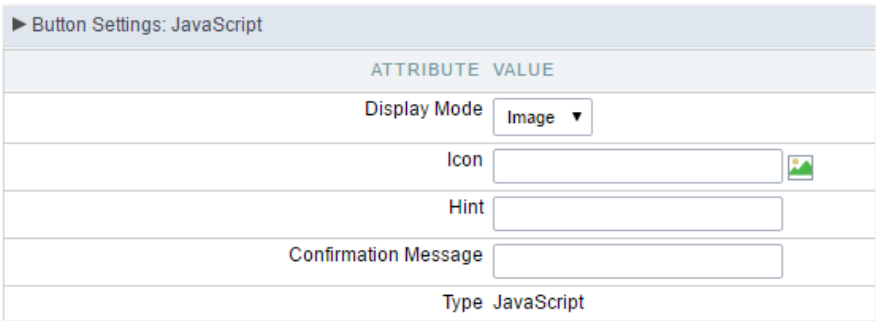
Button



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

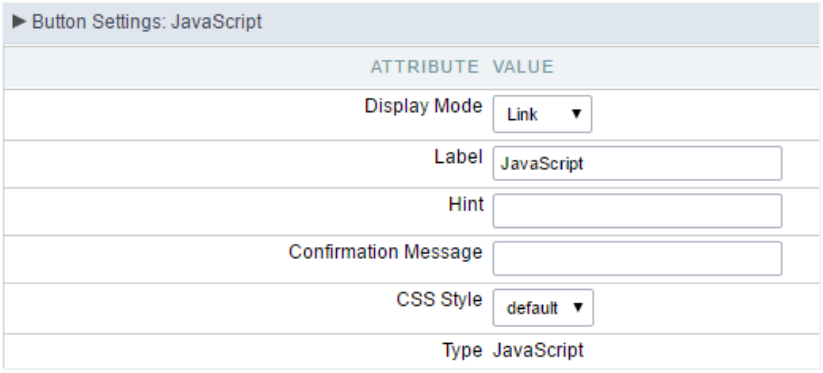
Image



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

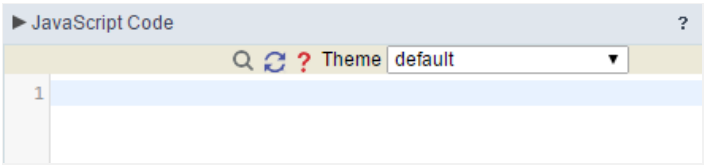
Link



Setting up Javascript Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Code Block



Javascript button coding block.

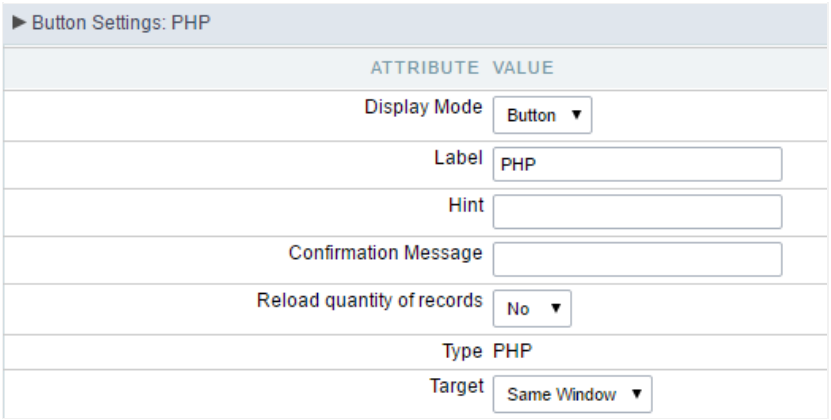
In this block, only JavaScript is accepted.

PHP

Display Mode

You can configure the display mode of the PHP button in Image, Button or Link.

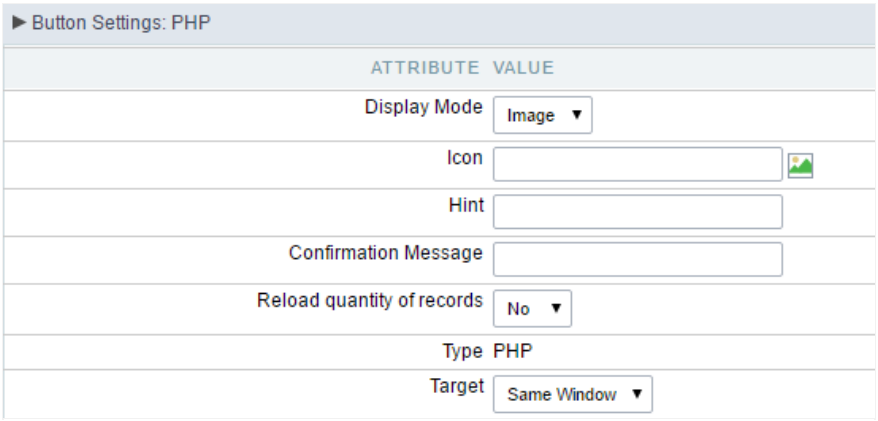
Button



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

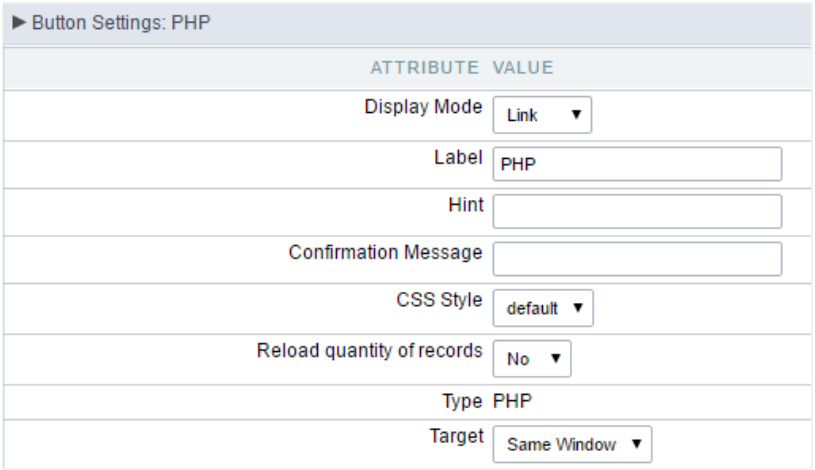
Image



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

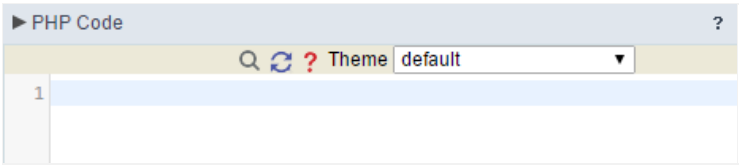
Link



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

Code Block



*Ajax button coding block. *

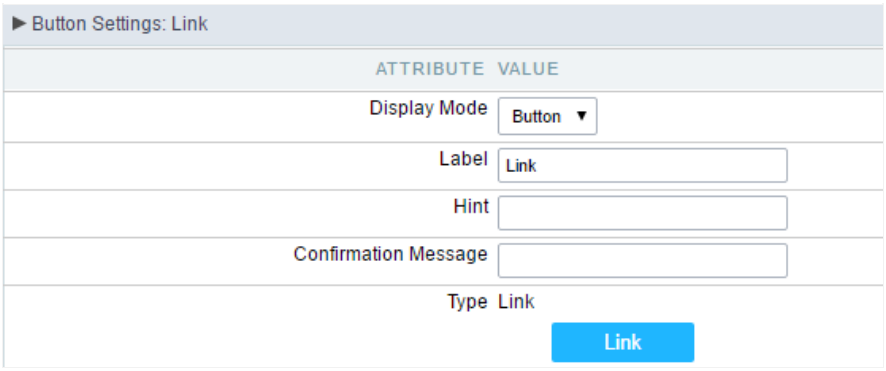
In this block, you can use macros, PHP code and JavaScript.

Link

Display Mode

You can configure the display mode of the link button in Image, Button or Link.

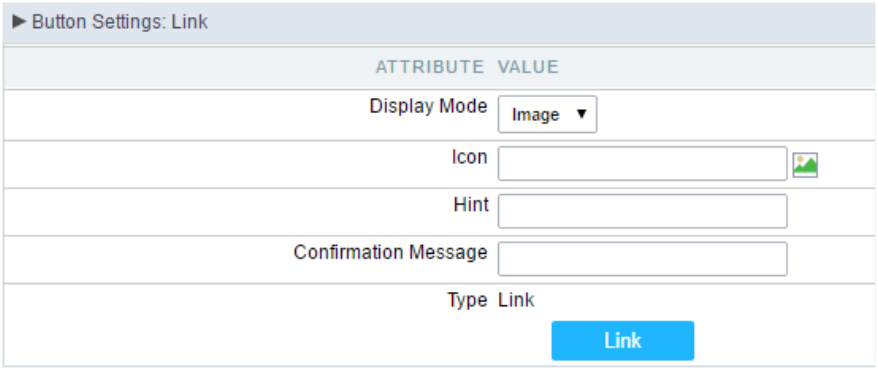
Button



Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Image

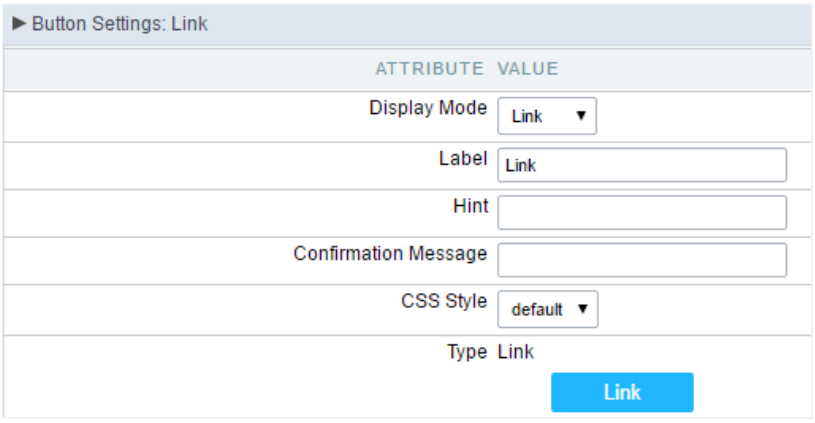


Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.

Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Link

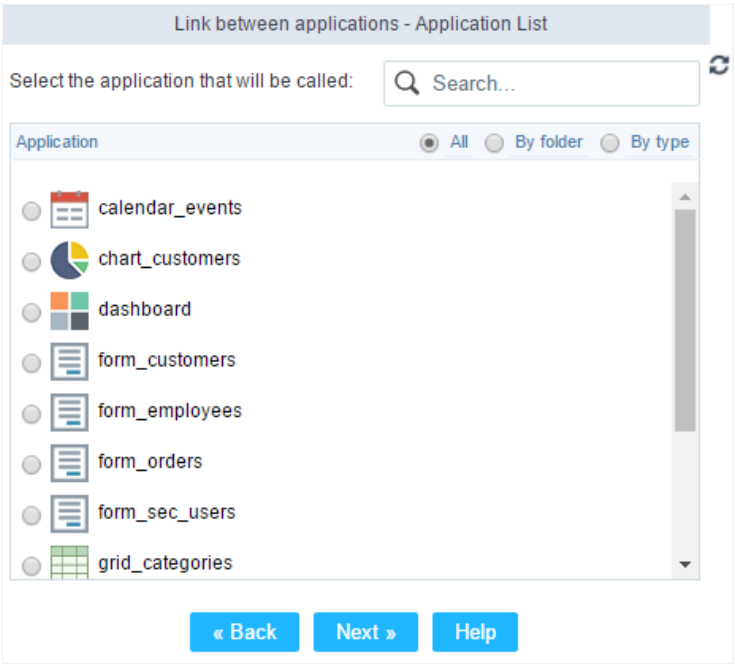


Setting up link Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Setting up the Link

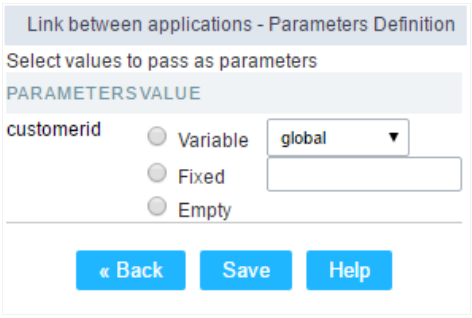
- Selecting the applications



Choosing the application for the button link.

You should select an application to be called from the button link.

- Link Parameters



Choosing the parameters for the button link.

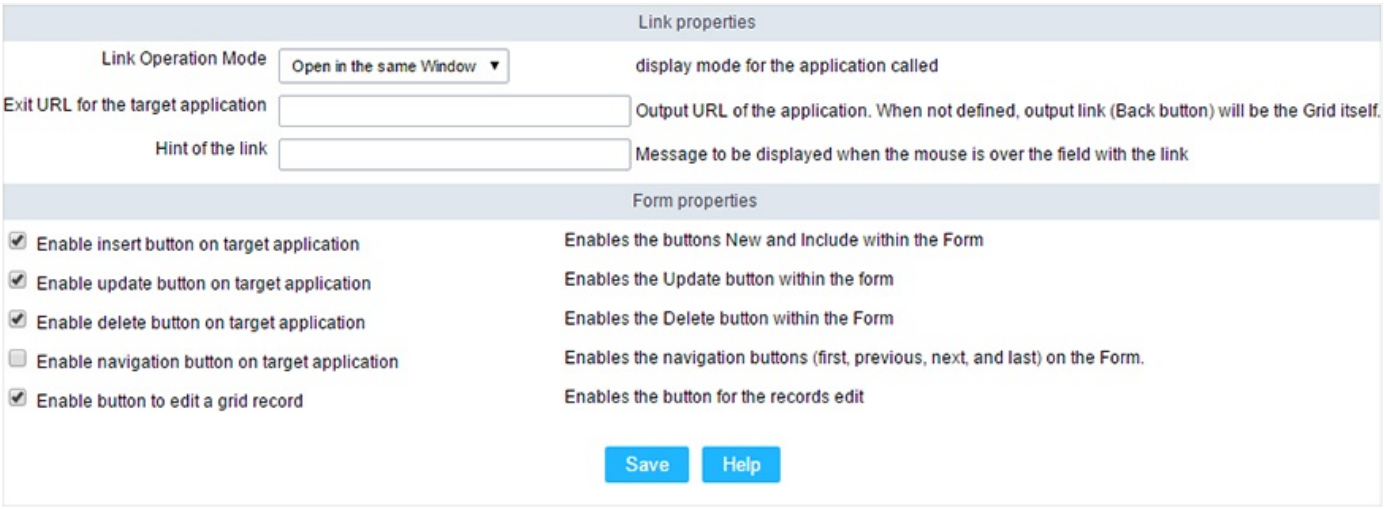
Field Allows you to use an existing field from the current application as a parameter for the link.

Variable Allows you to use a global variable from the current application as a parameter for the link.

Fixed Allows you to inform a fixed value as a parameter for the link.

Empty No value will be passed as a parameter for the link.

- Link Properties (Grid)



Configuring the properties for the link button when the destined application is a Grid.

Link Operation Mode How the link will open (Open in the same Window, Open in another Window, Modal).

Exit URL for the target application URL or an application that be redirect to when exiting the Grid application.

Initial Mode Allows you to define the initial mode of the grid application (Search or Grid).

Number of Lines Allows you to define the amount of lines displayed in the Grid.

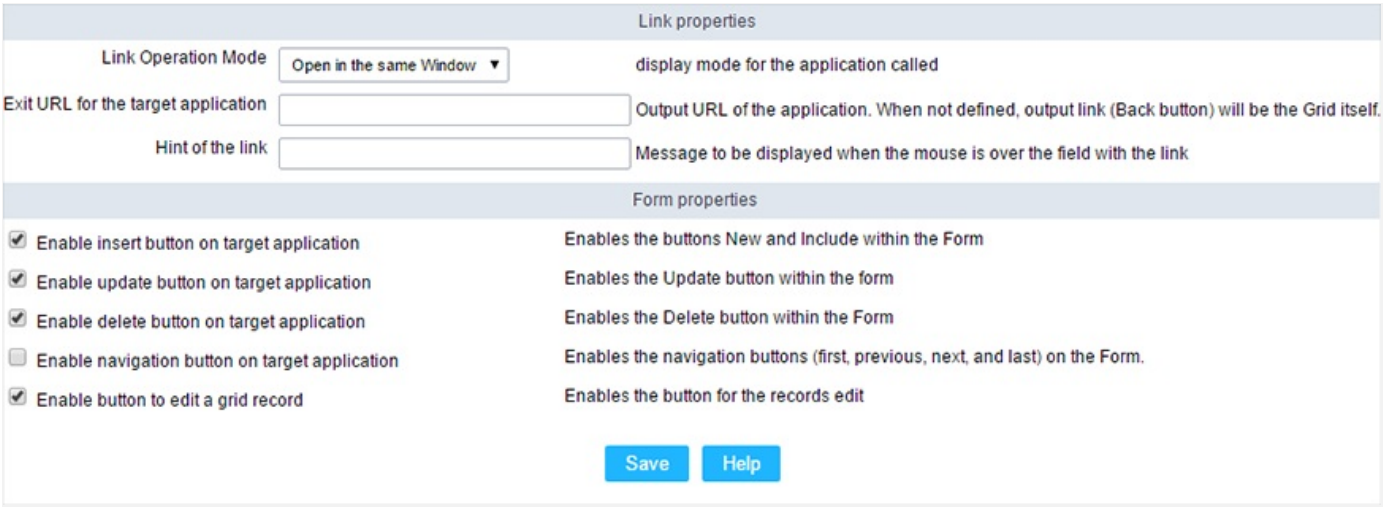
Number of Columns Allows you to define the amount of columns displayed in the Grid.

Paging Enable the paging in the Grid.

Display Header Enable the Grid Header.

Active Navigation Buttons Enable the navigation button (First, Back, Next and Last) in the Grid.

- Link Properties (Form)



Configuring the properties for the link button when the destined application is a Form.

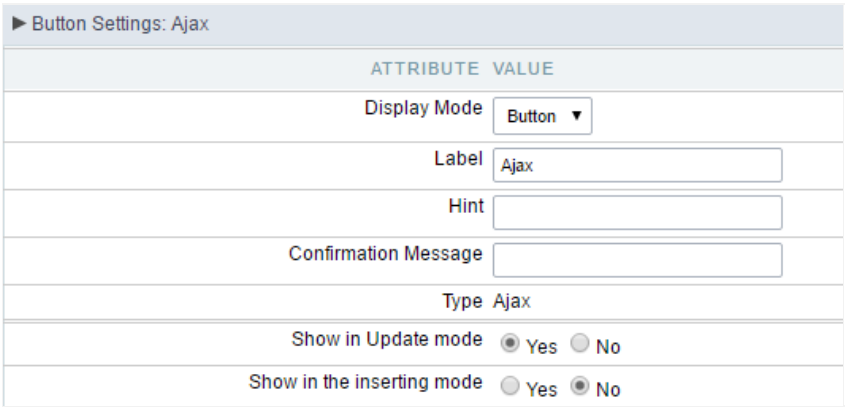
Link Operation Mode	How the link will open (Open in the same Window, Open in another Window, Modal).
Exit URL for the target application	URL or an application that be redirect to when exiting the Form application.
Enable insert button on target application	Enable the “New” button in the Form Application.
Enable update button on target application	Enable the “Update” button in the Form Application.
Enable delete button on target application	Enable the “Delete” button in the Form Application.
Enable navigation button on target application	Enable the navigation button (First, Back, Next and Last) in the Form.
Enable button to edit a grid record	Enable the buttons that allow you to edit the records of a Grid

Ajax

Display Mode

You can configure the display mode of the Ajax button in Image, Button or Link.

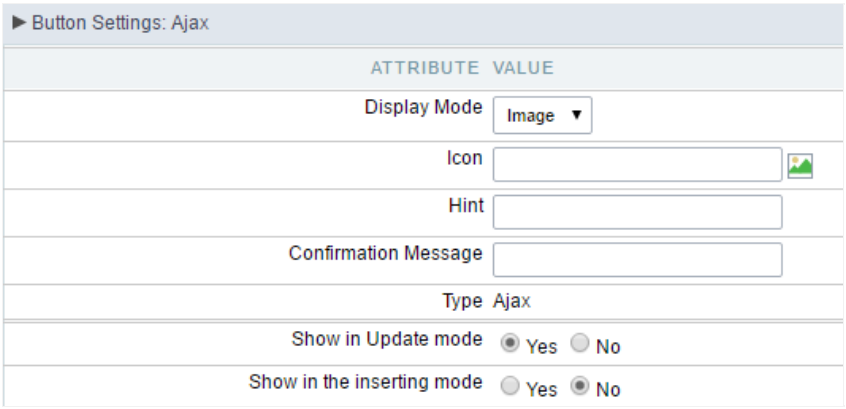
Button



Setting up Ajax Button.

Display Mode	You can select the display mode for the Ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

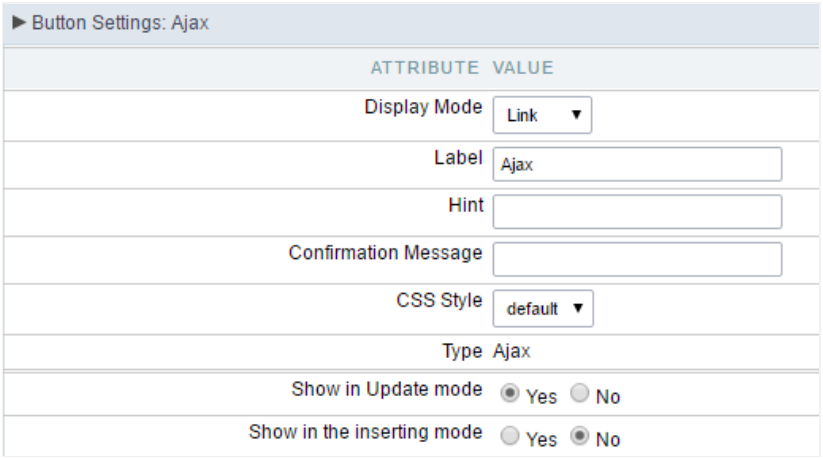
Image



Setting up Ajax Button.

Display Mode	You can select the display mode for the Ajax button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

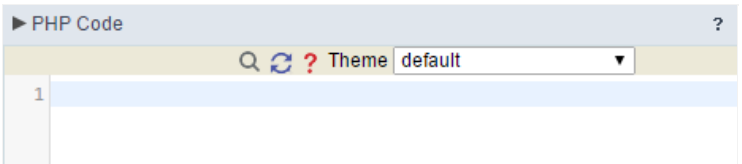
Link



Setting up Ajax Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Code Block



Ajax button coding block.

In this block, you can use macros, Ajax code and JavaScript.

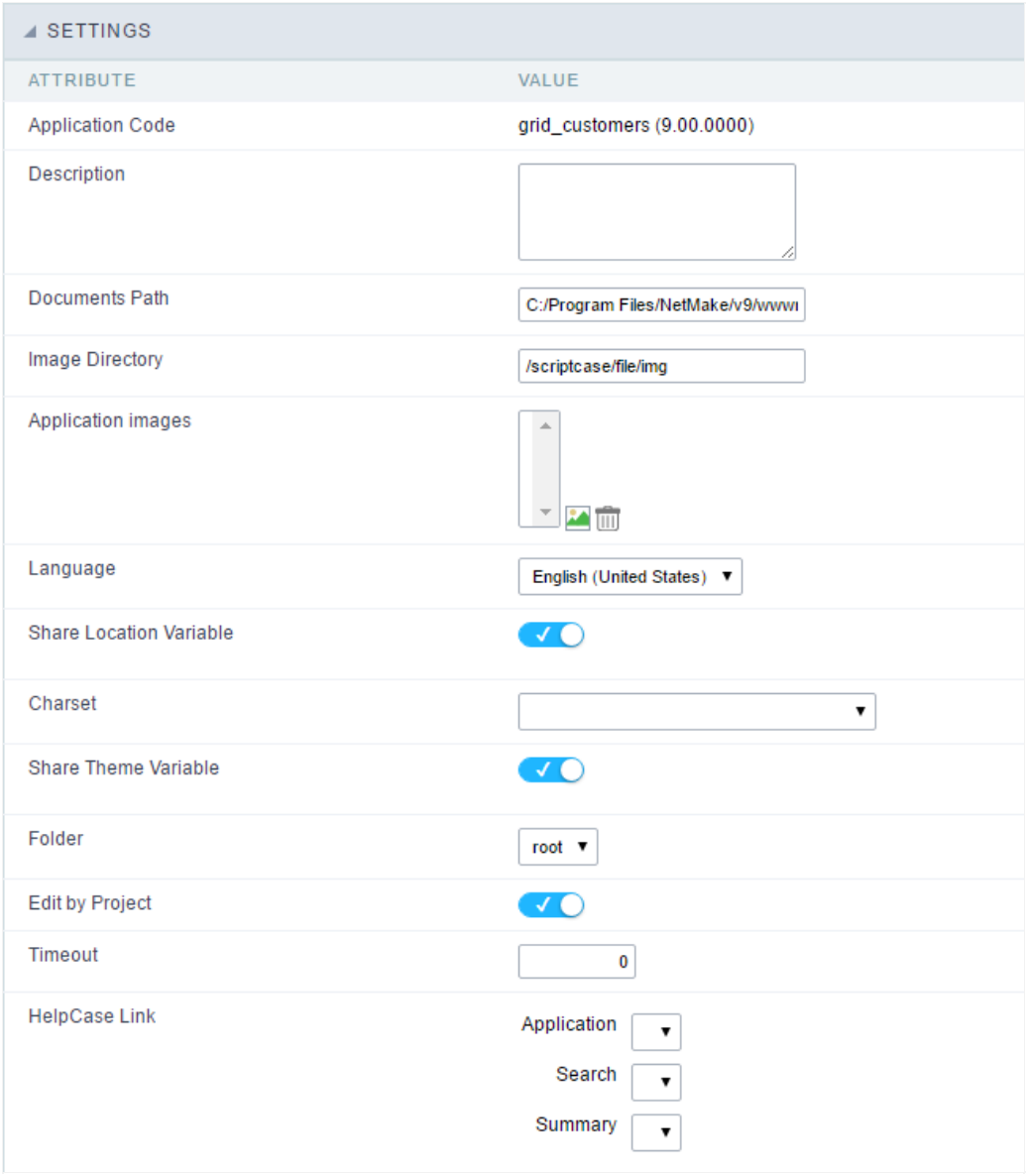
Related Links 🔗

Related Videos ▷

Application Settings (Control)

Settings

With this interface, you can set the common attributes of the app.

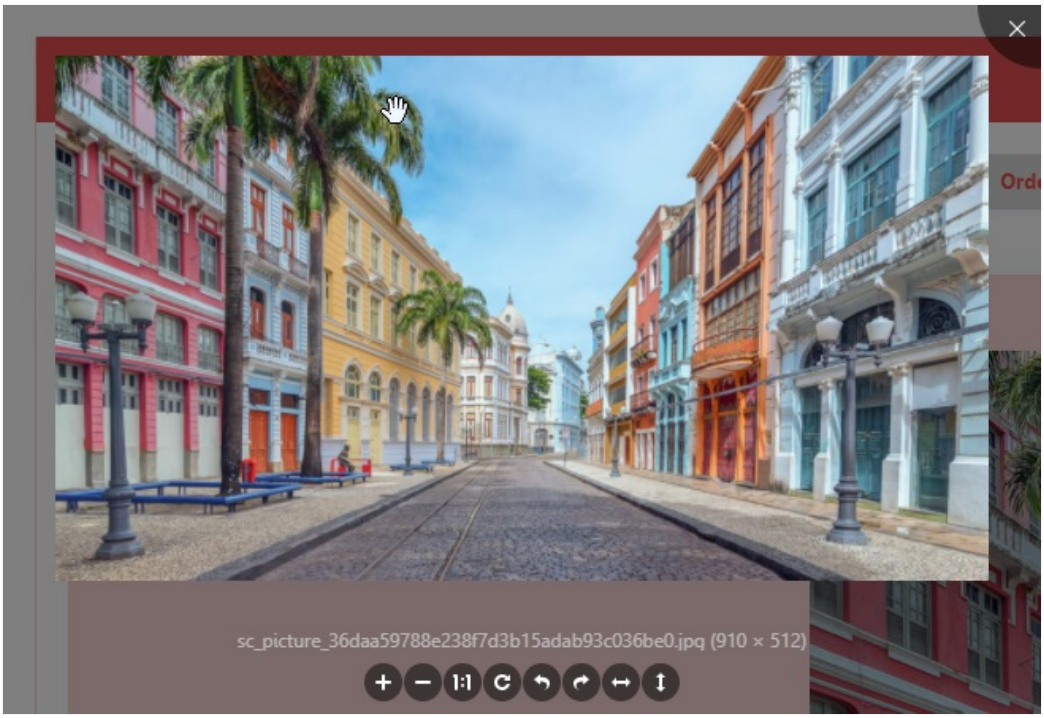


Application Settings Interface

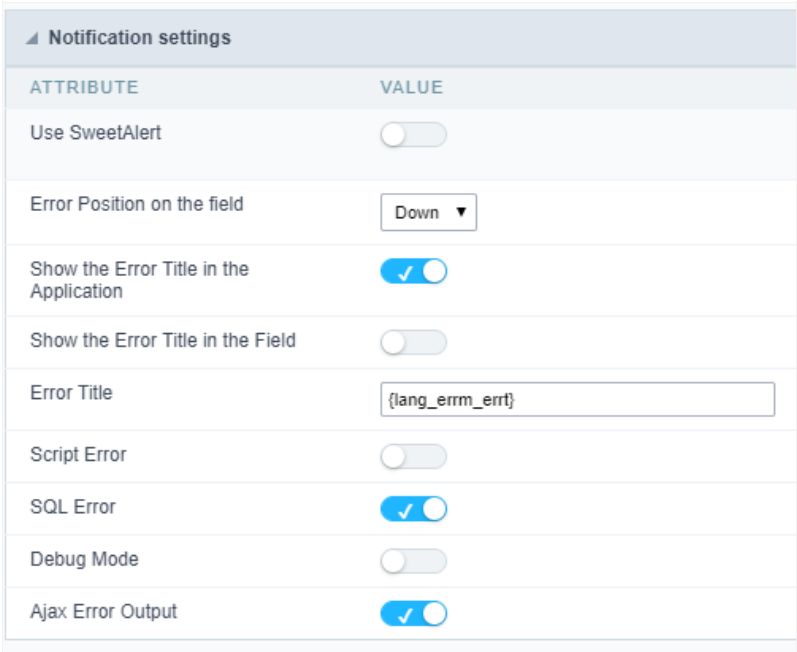
- **Attributes**
  - **Application Code** : It is the name that defines an application. An app can be renamed at the List of Application.
  - **Description** : This field contains a brief description of the application objectives.
  - **Documents Path** : The absolute path to store uploaded documents in the application.
  - **Image Directory** : The filesystem directory to store the application images.
  - **Application images** : Import images into the application to allows using them in the application.
  - **Language** : Set the default language of the application. Display all the application hints and messages in the selected language.
  - **Share Location Variable** : Define if the app shares the regional settings with other applications through a session variable.
  - **Charset** : Define a specific charset to use in the application.
  - **Share Theme Variable** : Define if the app shares the Theme settings with other applications through a session variable.
  - **Folder** : Define the project folder that contains the app.

- **Edit by Project** : Define if other project developers can edit the application.
- **Timeout** : Set the session runtime timeout in seconds. If the value is Zero, it assumes the default timeout of the PHP.
- **HelpCase Link** : It allows to associate a HelpCase file with the application.

- **Image Viewer**: Activates the image viewer features in the running application. By clicking on the image the user can drag, rotate, invert, and enlarge the selected image.



Notification Settings



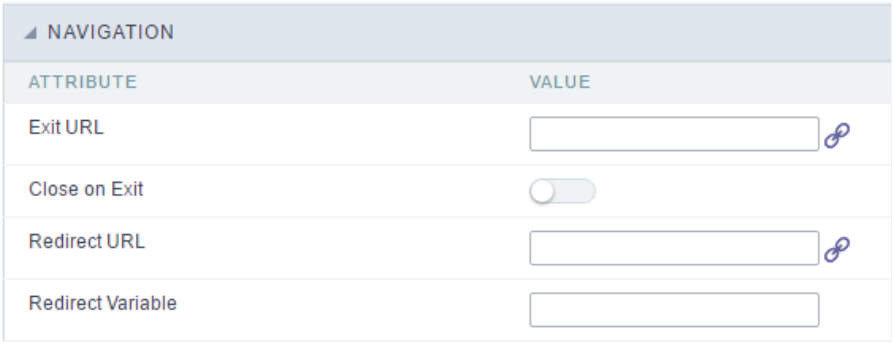
- **Use SweetAlert**: Use the SweetAlert to display messages from the application. When this option is active, it will replace the browser's "confirm" and "alert".
- **SweetAlert position using Toast** : The position to display error messages on the application.

- **Error Position on the field** : The position to display error messages when criticizing the field.
- **Show the Error Title in the Application** : Define to display the title line of the error message or not.
- **Show the Error Title in the Field** : Define to display the title line of the error message in the field or not.

- **Script Error** : Allows displaying the line code where there is an error..
- **SQL Error** : Allows displaying the SQL statement if it got an error.
- **Debug Mode** : Runs the application in Debug mode, showing all SQL statements the application is executing.
- **Ajax Error Output** : Enables the Ajax alert for debugging errors.

Navigation

This interface allows defining the navigating behavior of the application



Navigation Interface.

Exit URL

URL to where the user goes when he clicks on the “exit” button.

Close on Exit

Close the browser window when the user clicks on the “exit” button.

Redirect URL

Redirect to another URL in case there aren’t any global variables available.

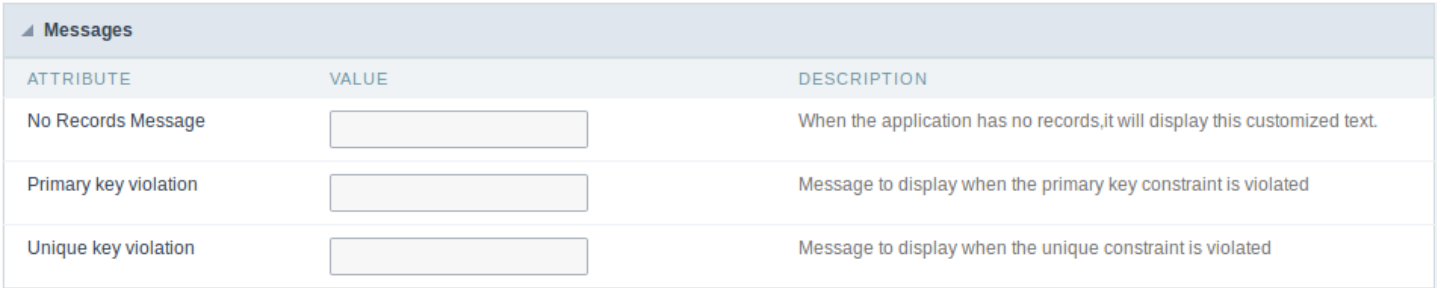
Redirect Variable

Creates a variable with the application name and sends it to the redirected application.

Messages

On this screen, you can redefine the default application messages to the end-user by customizing the validation messages for Insert, Update, and Delete. You can also define messages for SQL errors and the confirmation outputs.

Messages



No Records Message

The message displayed when the application has no records.

Primary key violation

The message displayed when there is a violation in the database's primary key constraint.

Unique key violation

The message displayed when there is a violation in the database's unique constraint.

Insertion Messages

Insertion messages

ATTRIBUTE	VALUE	DESCRIPTION
Message After Insert		Message to display after inserting a record.
Message to Confirm Insert		Message to display to confirm inserting of a record.

Message After Insert

The message displayed when inserting a new record.

Message to Confirm Insert

The message displayed if the end-user wants to confirm the inserting of a new record.

Update Messages

Update messages

ATTRIBUTE	VALUE	DESCRIPTION
Message After Update		Message to display after updating a record.
Message to Confirm Update		Message to display on the update confirmation

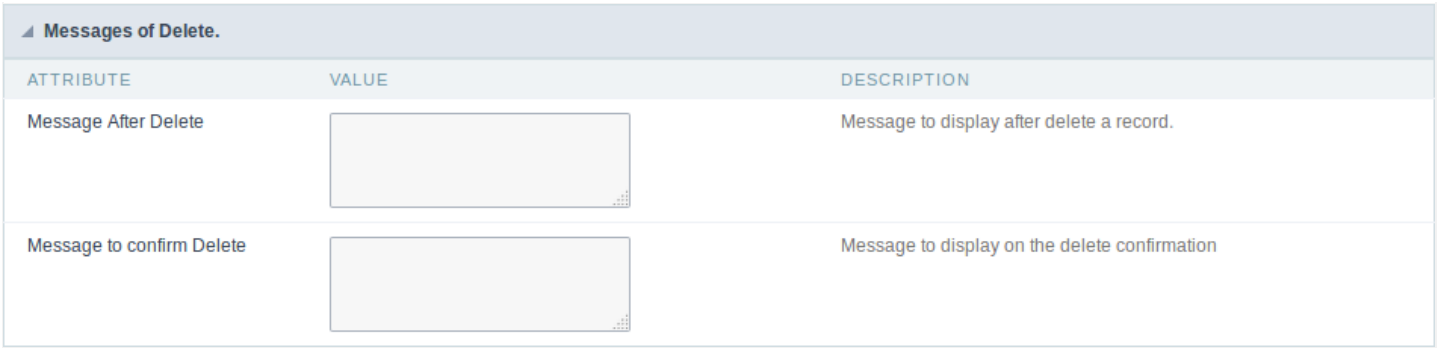
Message After Update

The message displayed when updating a record.

Message to Confirm Update

The message displayed if the end-user wants to confirm the changes of a record.

Messages of delete



Message After Delete

The message displayed when deleting a record.

Message to confirm Delete

Displays a customized message asking to confirm the record deletion.

If you make no changes, the application will use the default values for the messages. Those values can be configured in Locales -> Application Language.

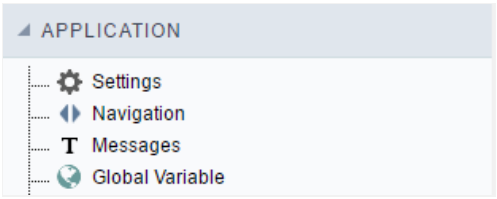
Global variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target="_blank"} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



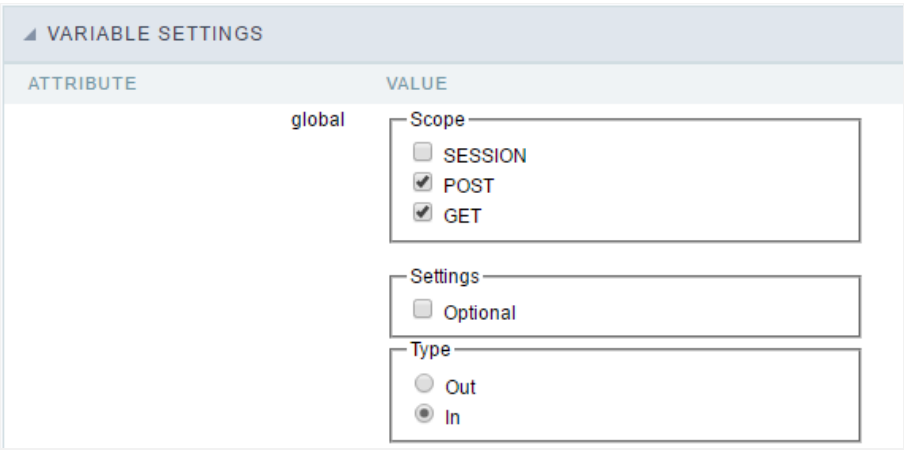
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

Indicates where the application is using the variable.

Related Links 🔗

Related Videos ▷

Control Links Overview

This feature allows the developer to create links between applications of the same project, expanding the integration of applications. All link options are grouped under the Application Links menu.

Application Links

In the first access to the menu, we can see the list of existing connections in the application.

> If the application does not have a configured link, the application list screen will be displayed with the message: **This application does not have any link. Click here to create one now.**

ID

Link identification ID.

Type

The type of link created, some links such as Edit Link allow only one link per application. In this case, the developer will be able to check the connection types that already exist in the application

Target Application

This info show the target application name.

Actions

This column has edit options for the links.

Properties

It allows accessing the binding properties where it is possible to configure the binding behavior.

Link

Displays the links screen, where it is possible to configure the link that was made with the application informed in the target application column. In this option it is possible to change the parameter passed in the connection as well as the target application.

Delete

Permanently deletes the connection in the applications.

> Through the option Restore Applications, it is possible to get a previous version of the application, making it possible to recover the link deleted.

Links Type

The Form application has the following links options

> The HTML control has only the application binding available in the interface.

- Application Link - Allows you to link a control with any project application.
- Button Link: It allows creating a link through the buttons created by the developer in the query application, to any application in the project.

Control Applications Link Types x Target Application available to link

Check the links types of the control application and the target applications available for each link types.

Only **Capture link** have some restrictions in the target application use, allowing only to use of Grid application.

	Application Link	Button Link
Grid		
Chart		
[**Form**][anc_gv_form]{:target="_blank"}		
[**Control**][anc_gv_ctrl]{:target="_blank"}		
[**Search**][anc_gv_search]{:target="_blank"}		
Menu		
Tree Menu		
Tabs		
PDF Report		
Dashboard		
Blank		
[**Calendar**][anc_gv_calendar]{:target="_blank"}		

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Application link

Creating an Application Link

This type of link allows the developer to create a link from a grid to a form with the objective of editing the register of a grid's row.

In the link options, we will choose the **Application Link**. When we choose this option, edit a register from a Grid will be possible.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Open in an iframe:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a iframe in one of the four options available in the iframe settings.

Open in a parent:

When we use this option the target application will be displayed in the same window of our application, and the target application will have a back button so we can return to the previous application.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window.

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set these following options:

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

Related Links

- Login with language selection
- Contact Form template
- Registration form template
- Responsive login template
- Control introduction
- Control overview

Related Videos

- Form overview

Control Button link Settings

Creating a Button Link

Allows the developer to create a link where the call to the other application will be done through a button.

In the type of links options, we will choose the **Button Link**. Choosing this option it will be possible to create a link to any other application.

List of applications

After selecting this option, The list of applications to what you want to create a link will be displayed.

This screen can be viewed in the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open the application in a tab inside the menu:

Choosing the option **Yes**, the application will be open in a **menu tab**, when executed from a menu application, instead of opened in another browser window.

Choosing the option **Yes**, there are also some other settings to be displayed:

Title to the tab:

This option allows the developer to set a title to the tab that will be opened when used in a Menu application.

Hint to the tab:

This option allows the developer to set a message to be displayed when the mouse cursor is over the tab Menu.

Active tab icon:

This option allows the developer to set an icon to be displayed in the tab when used in a menu application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Height:

Allows to set the Modal's height.

Width:

Allows to set the Modal's width.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Related Links

- Login with language selection
- n template
- form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

Related Videos

Control Edit Link Configuration

In the Link Folder of the Application Menu (Image Below) are displayed the links existing in the application and also the item New Link. When clicking on the existing link it is displayed the screen below that allows to manage the links.

Editing Links.

Actions

Properties	Change the link’s behavior, position, and how the link opens.
Link	Change the application that’s being called in the link their parameters.
Delete	Remove the existing link.

Related Links

- Login with language selection
- n template
- form template
- Responsive login template
- Control introduction
- Control overview
- Form overview

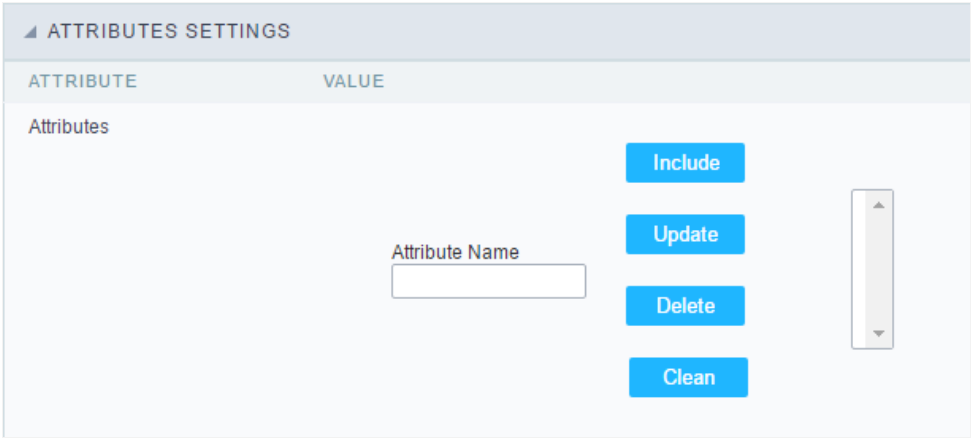
Related Videos

Control Programming

ScriptCase has incorporated the concept of Object Oriented programming, using attributes, resources, methods and libraries. It is possible to create your own business rules in applications, and by using these concepts you can reap huge rewards in terms of better organization and improved development.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

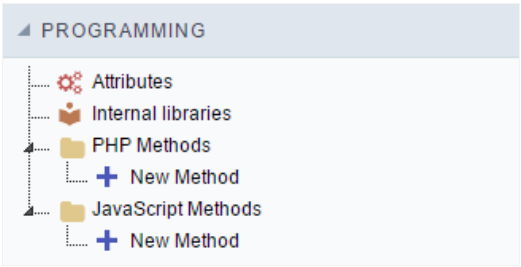
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

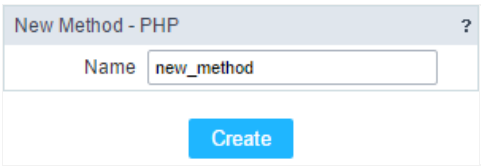
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

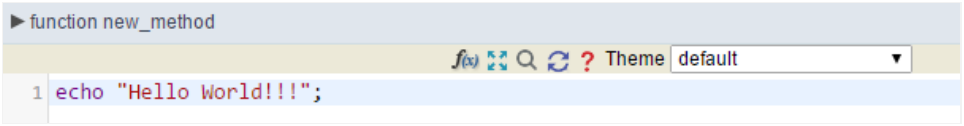


- Give the method a name and click Create. As in the image below.

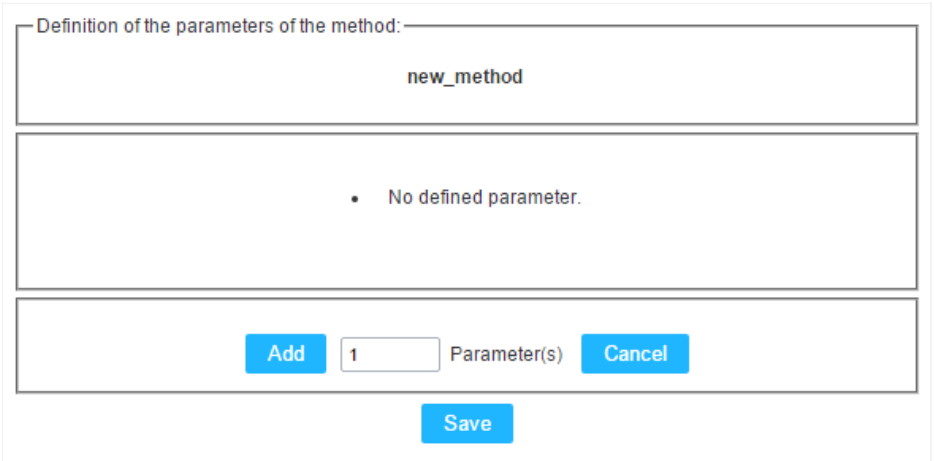
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



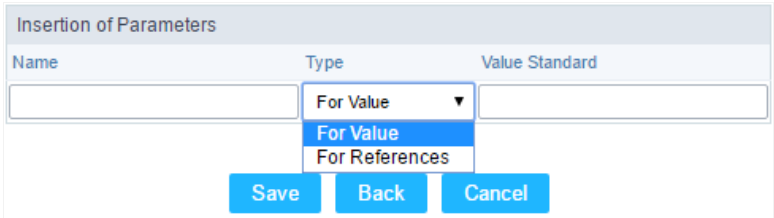
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

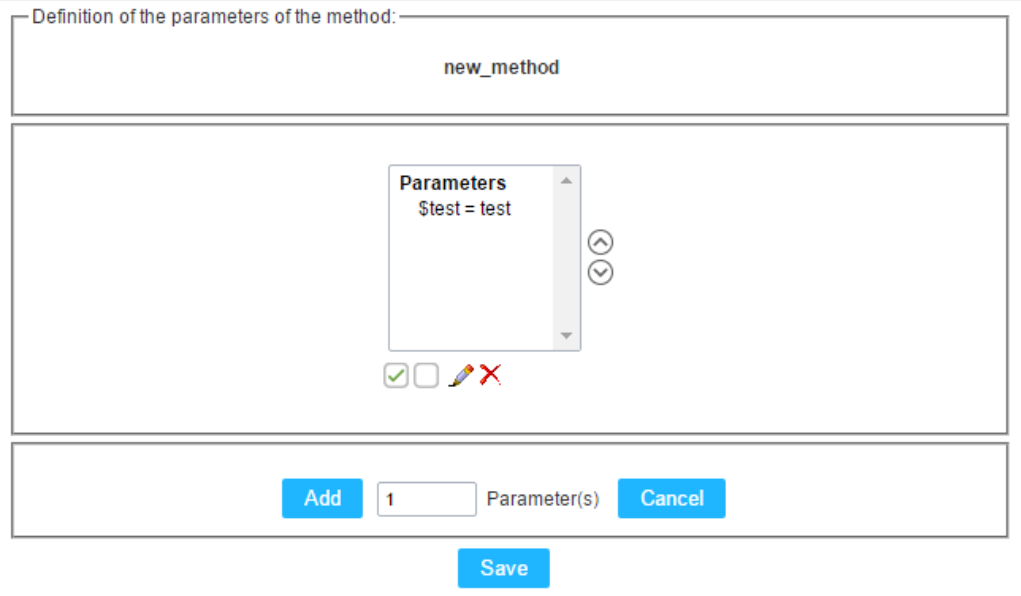


- **Name** : Type in the variable's name.
- **Type** : Selecting the type of variables: For Value or For Reference.

- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:

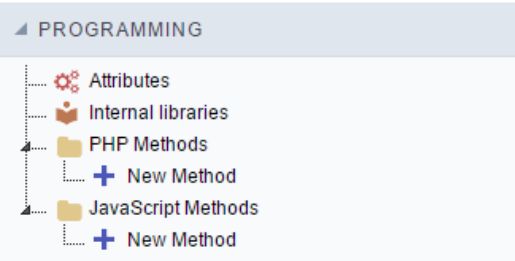


- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

JavaScript Methods

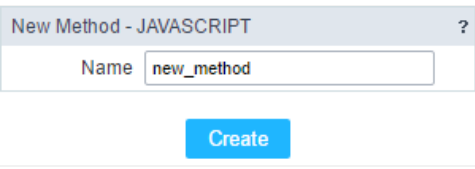
Methods are function or procedures declared by the developer, that helps when appling the routines. Using methods in the applications allows to reuse of the your code throughout the application, optimizing the development experience.

Creating a new method



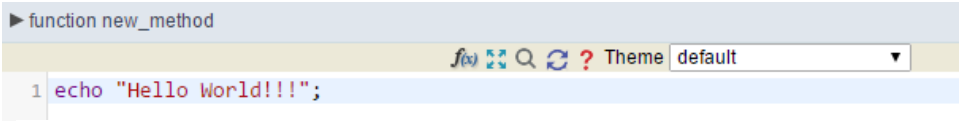
JavaScript method creation Interface

- Define a name for the method and click on Create. Like the image below.

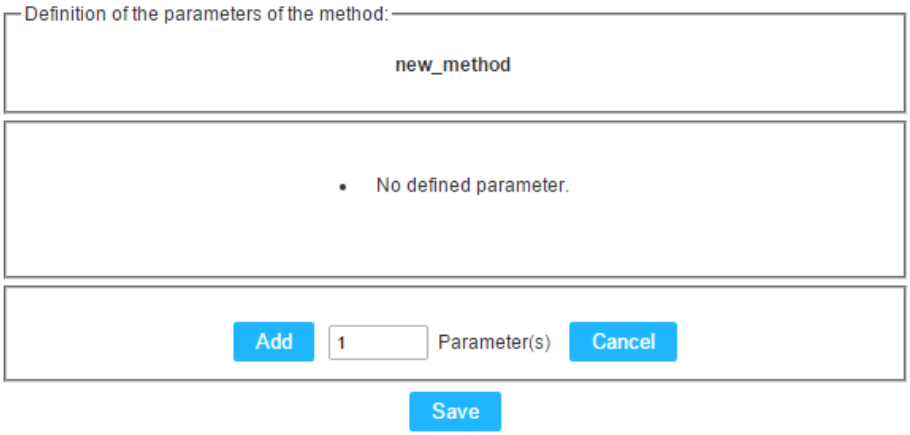


Include Method.

- Methods can receive parameters.

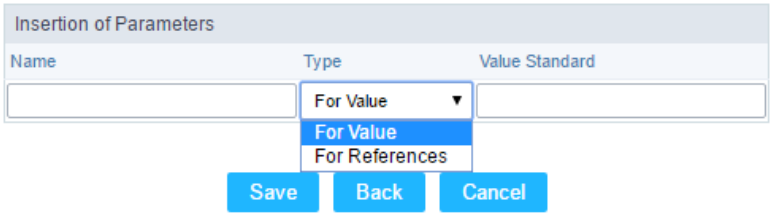


  - Add the amount of variables:



JavaScript method parameter definition

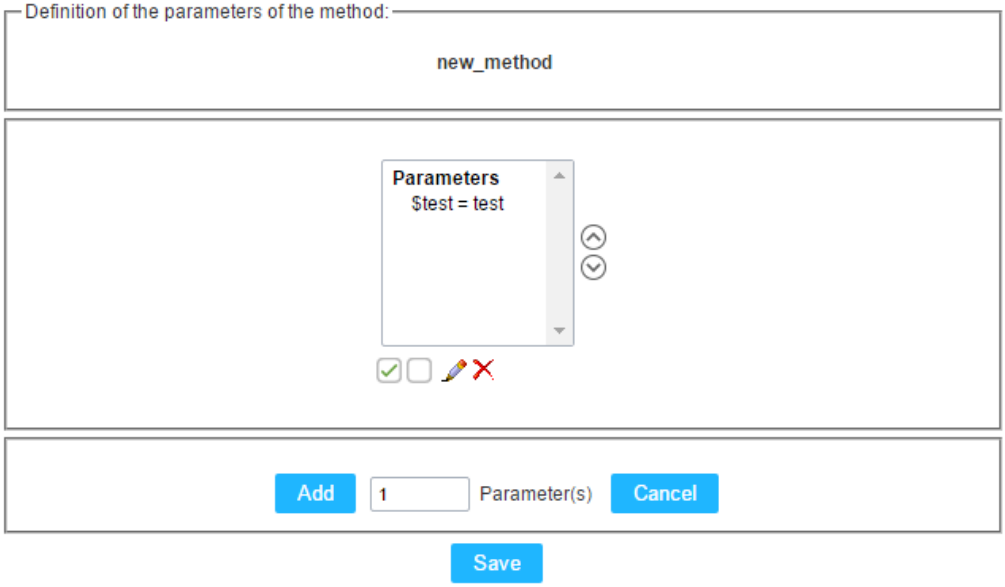
  - Defining the variables:



Defining the variables

    - **Name** : Type in the variable's name.
    - **Type** : Selecting the type of variables: For Value or For Reference.
    - **Value Standard** : The parameter's value used to initialize when calling the method.
      References allows to create a second name for a variable that you can use to read and modify the original variable's information.

  - Editing a parameter:



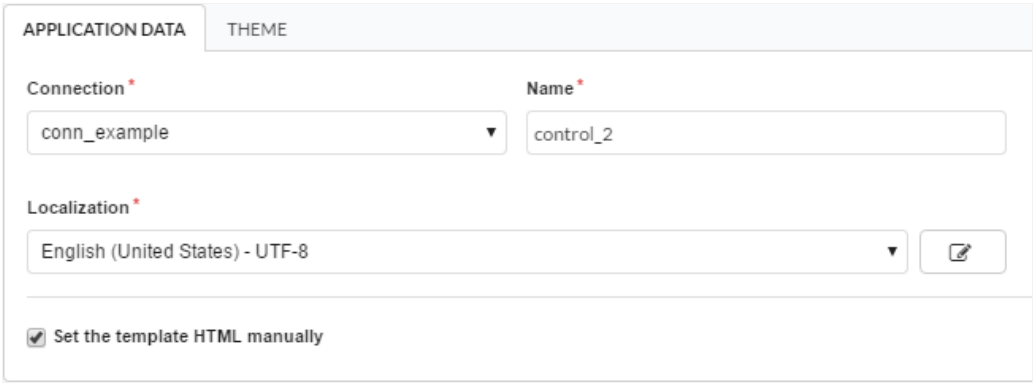
    - : Checking all the parameters in the parameter list.
    - : Uncheck all the parameters in the parameter list.
    - : Edit the selected parameter of the list.
    - : Deletes the selected variable of the list.

Related Links

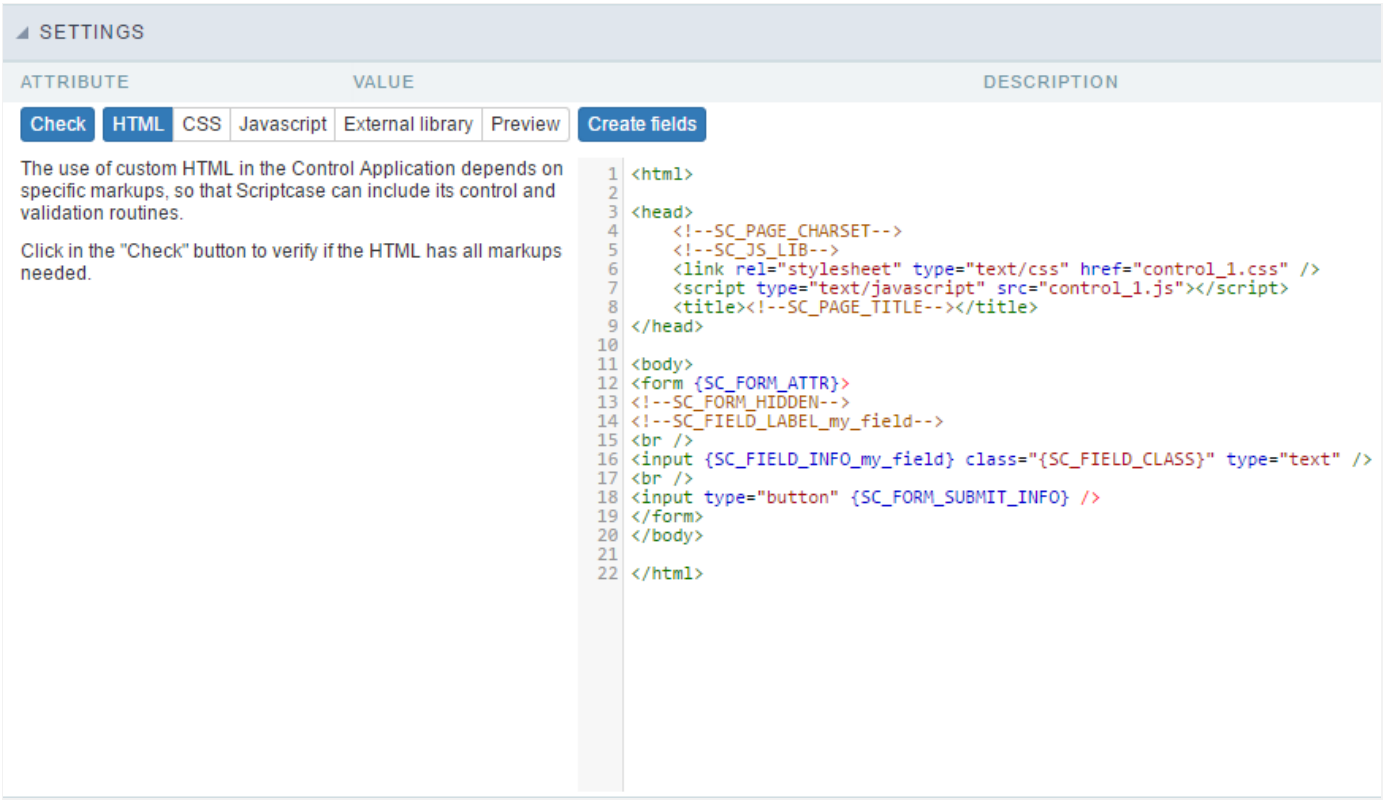
Related Videos

User HTML

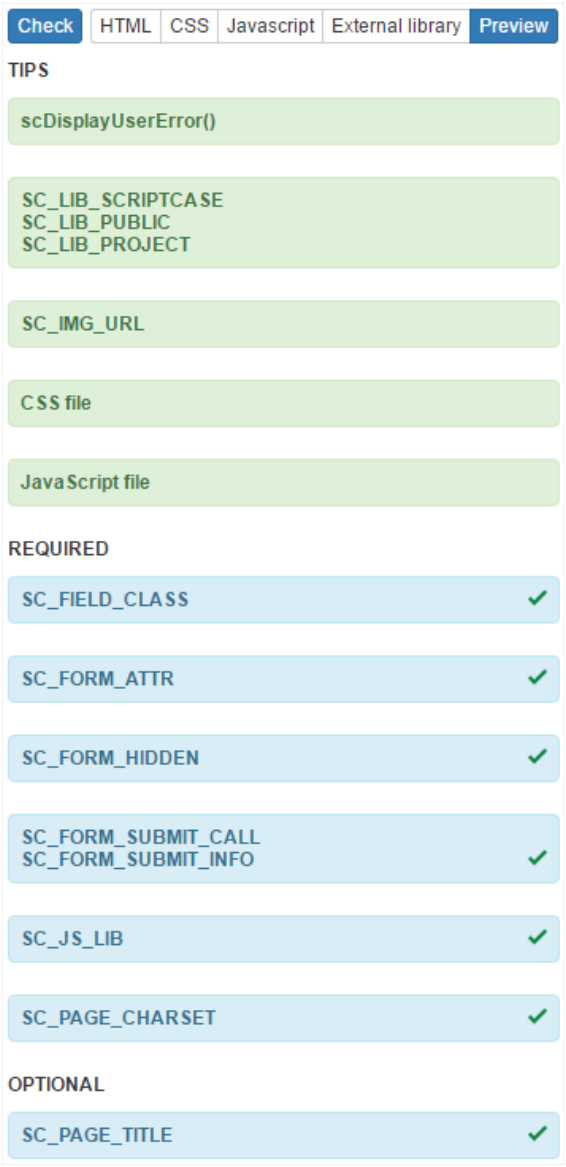
This option is available in the Control application if the item **Set the template HTML manually** template is selected when creating the application.



To use the custom **HTML** in Control, depends on specific **Markups** so that the Scriptcase can include its control and validation routines.



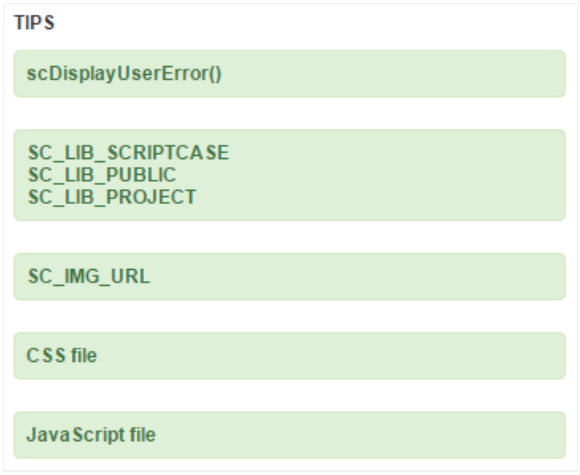
Check



Clicking the **Check** button, it verifies the structure to confirm which **Markups** were used in the content of the **HTML**.

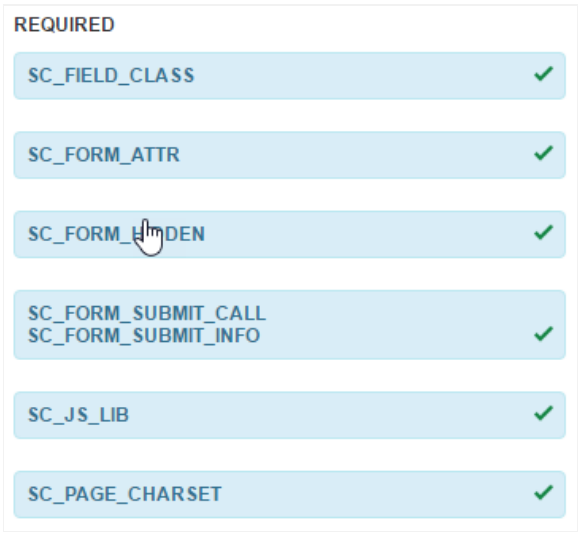
When clicking on the **Markups** of the **Check** item, a modal window is displayed with tips for using the same **Markup**.

- The **Markups** are divided in 3 different categories:
  - **TIPS**



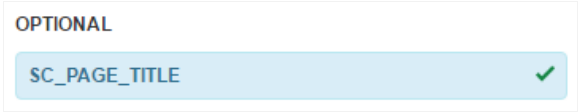
These **Markups** are not required, however, they are used to import files from libraries, images and other features.

- **REQUIRED**



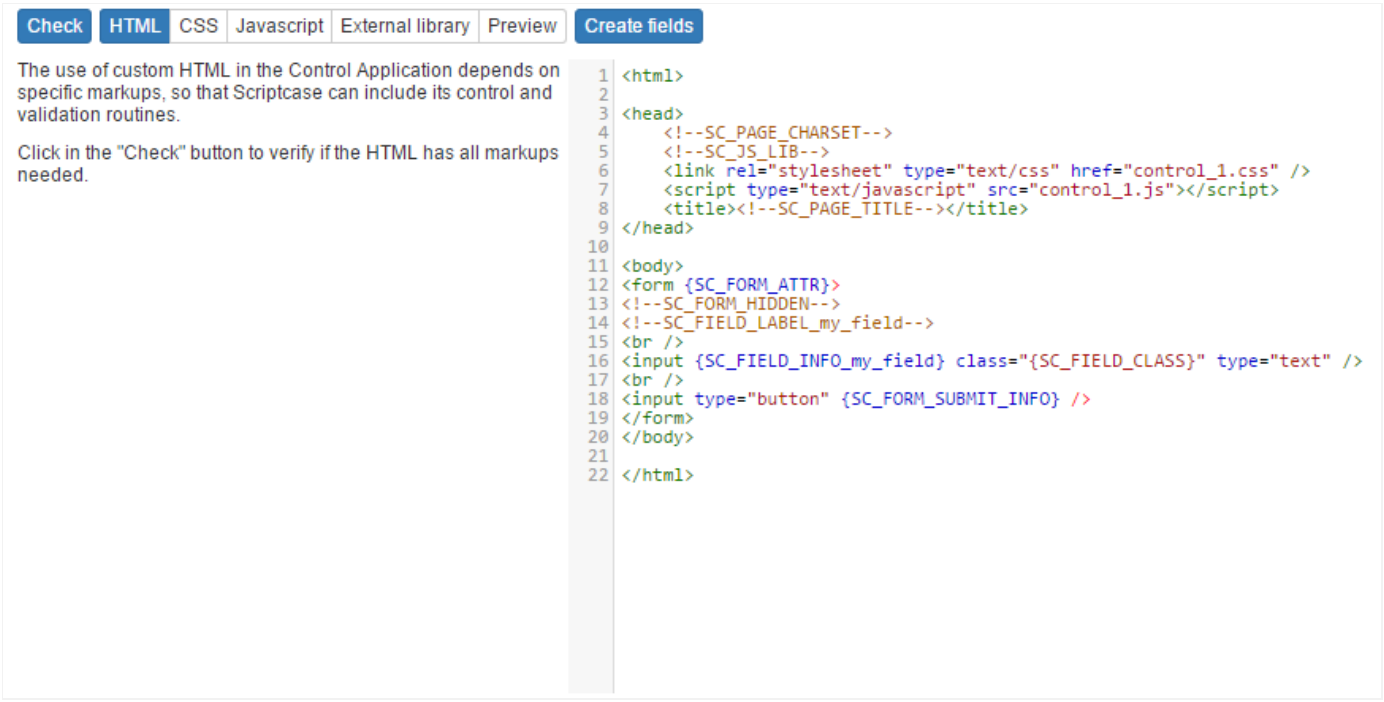
These **Markups** are of obligatory use in the item **HTML**. If they are not used, certain internal Scriptcase routines will not be executed.

- **Optional**



Like the category name, these **Markups** are optional, not influencing the correct execution of the application.

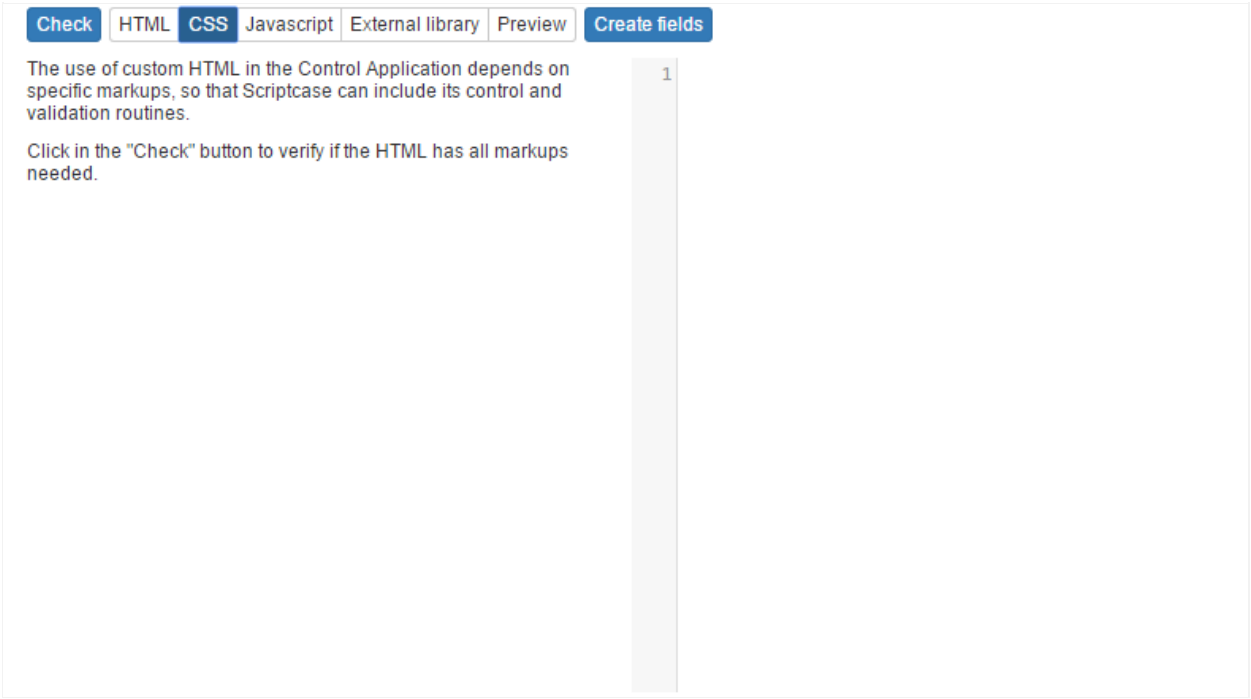
HTML



In this item, we can insert the desired HTML, remembering to use the required **Markups**.

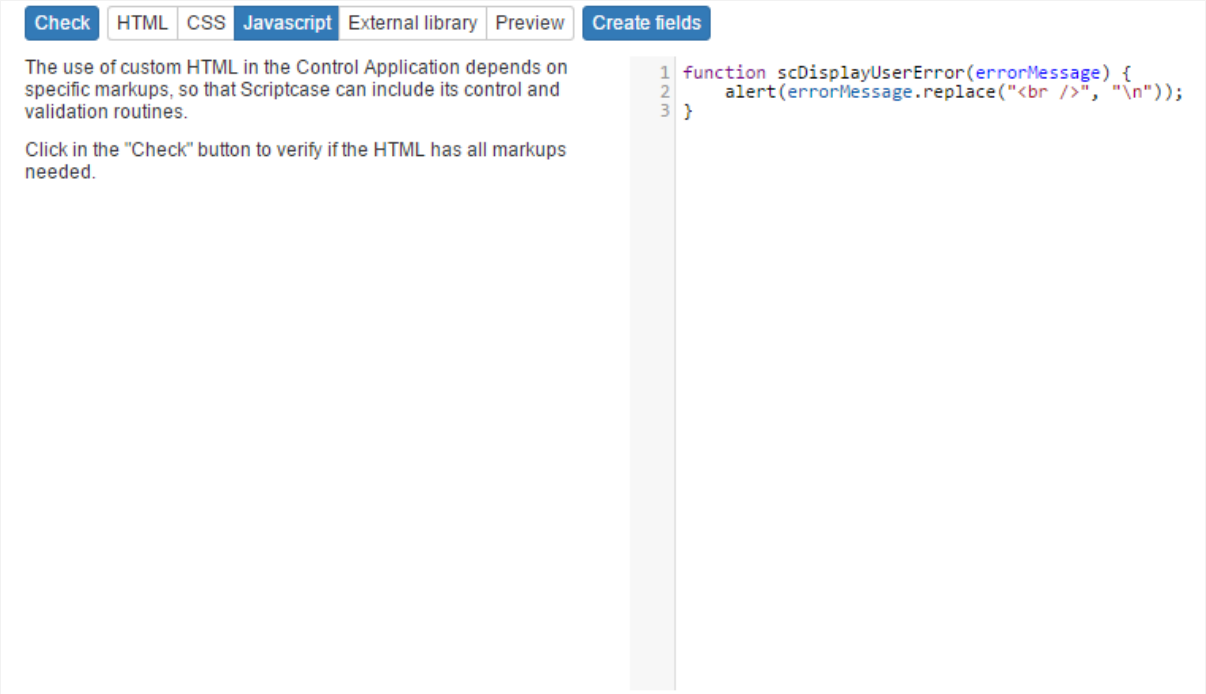
The **control.js** and **control.css** files refer to the contents of the **JavaScript** and **CSS** items respectively. By default, they are automatically created by the application, by default: **appName.js** and **appName.css**.

CSS



In this item, we can use our custom CSS to take effect in the **HTML** code used in the previous item.

JavaScript

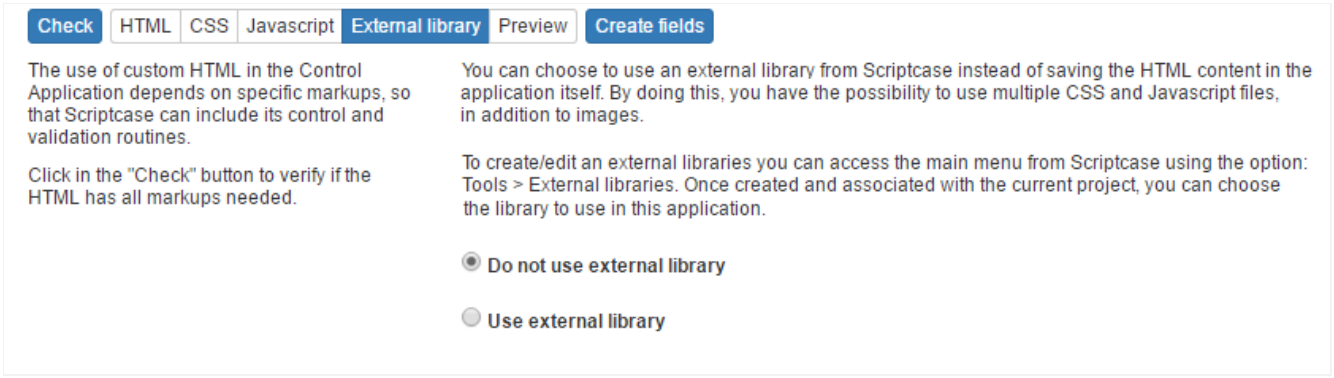


In this item, we can use our custom **JavaScript** to take effect in the code **HTML** used in the previous item.

External Libraries

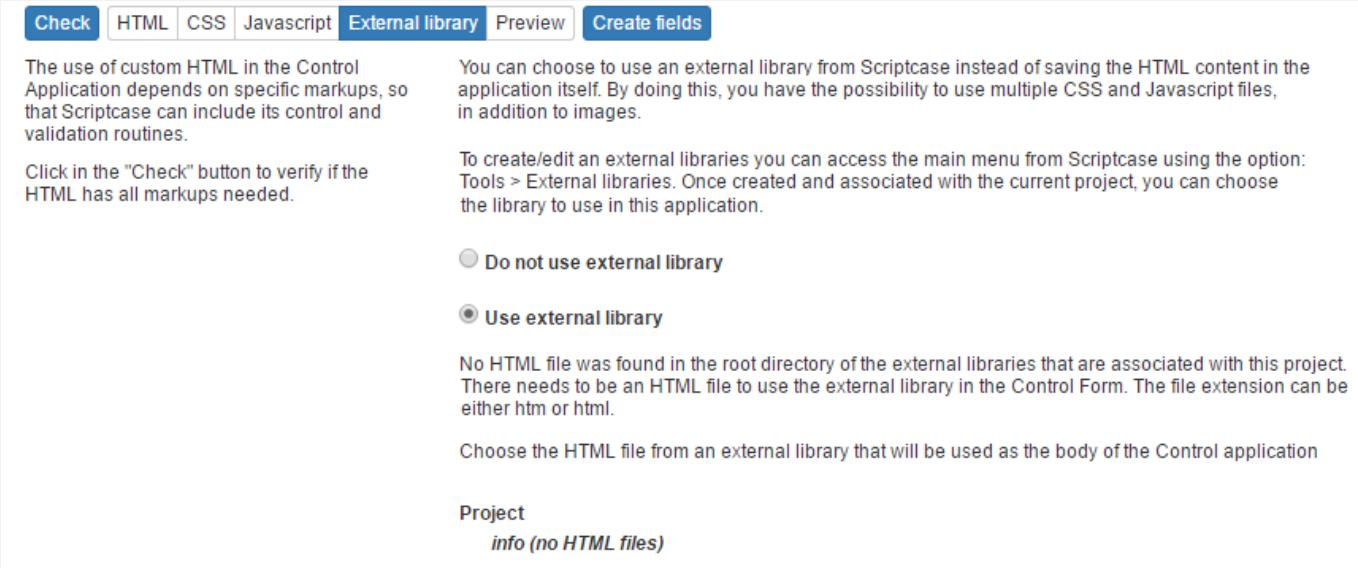
In this option, we can define if we will use the files **. HTML**, **. CSS** and **. JS** from an external library or if we will use the code **HTML**, **CSS** and **JavaScript** of the items of the application itself.

- Do not use external library



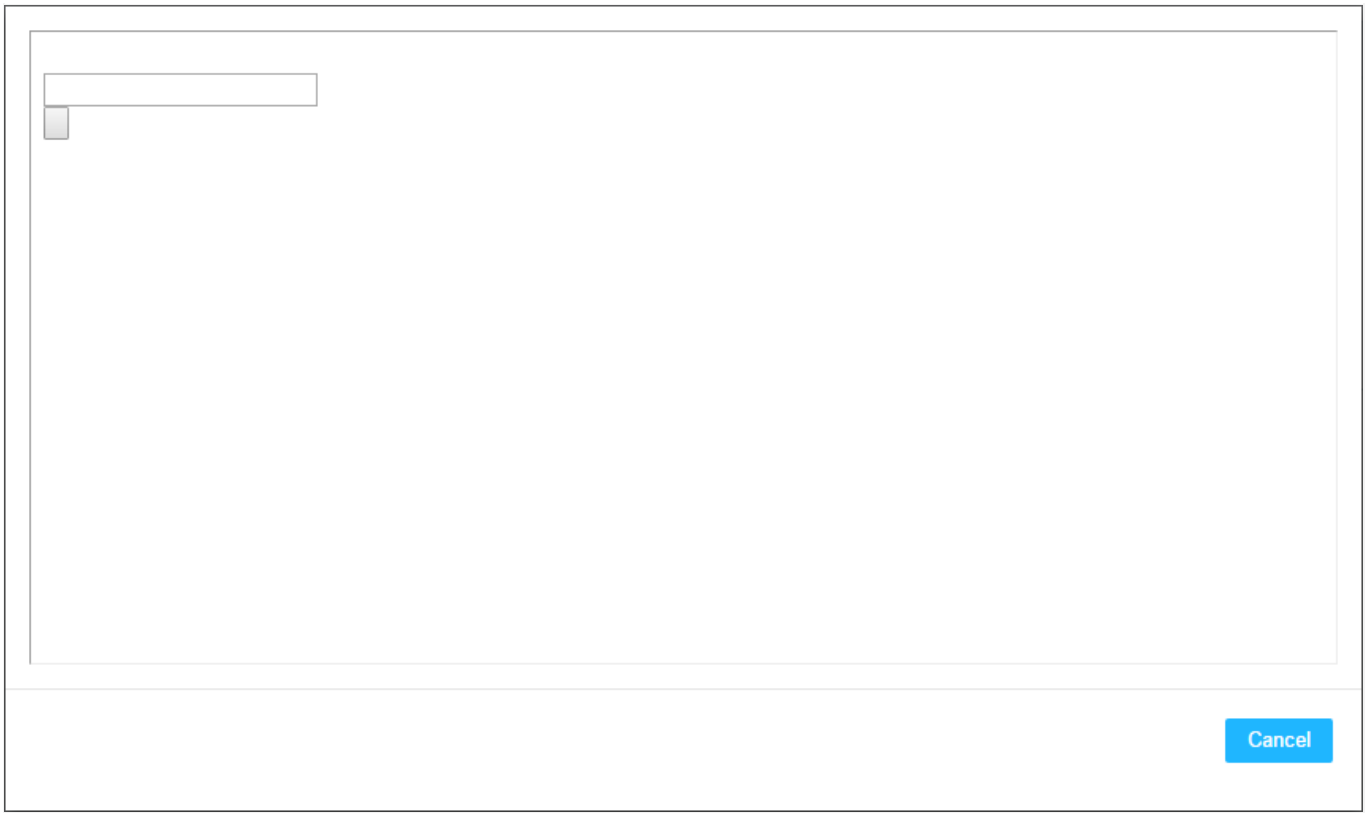
  This option defines that the **HTML**, **CSS**, and **JavaScript** code to be used, come from the current application's **HTML**, **CSS**, and **JavaScript** items.

- Use external library



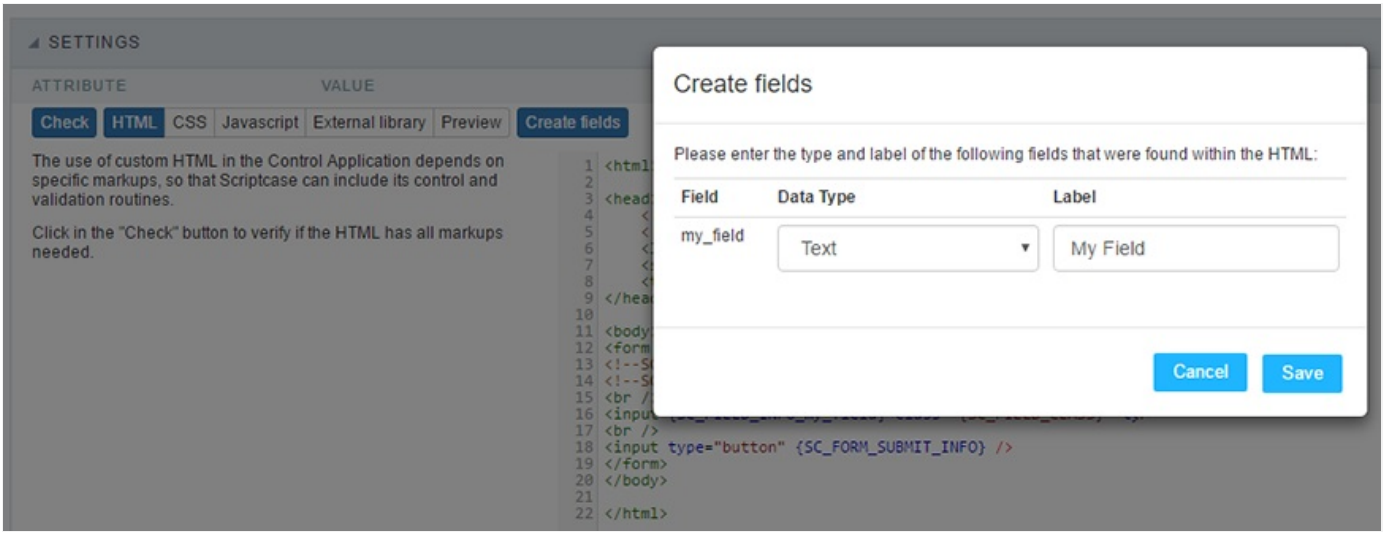
When selecting this option, we define the use of external libraries that contain the **HTMl**, **CSS** and **JavaScript** to be used by the application. The **HTML**, **CSS** and **JavaScript** items will be disabled.

Preview



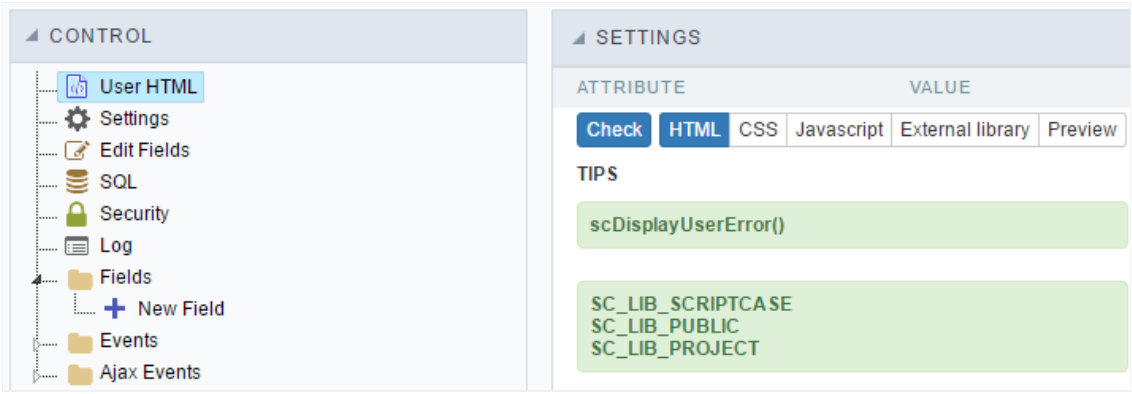
You will see how the application will appear after its generation.

Create Fields



If you use field Markup before the field is created in the menu item **Fields > New Field**, and the item (**Create Fields**) is pressed, a modal window is displayed to synchronize the fields, after confirmation, the fields will be displayed in the **Fields** menu item.

When adding fields from the menu item **Campos> New Field**, Scriptcase will automatically insert new mandatory **Markups** related to the newly created field to be used in the **HTML** item.



Related Links 🔗

Related Videos ▷

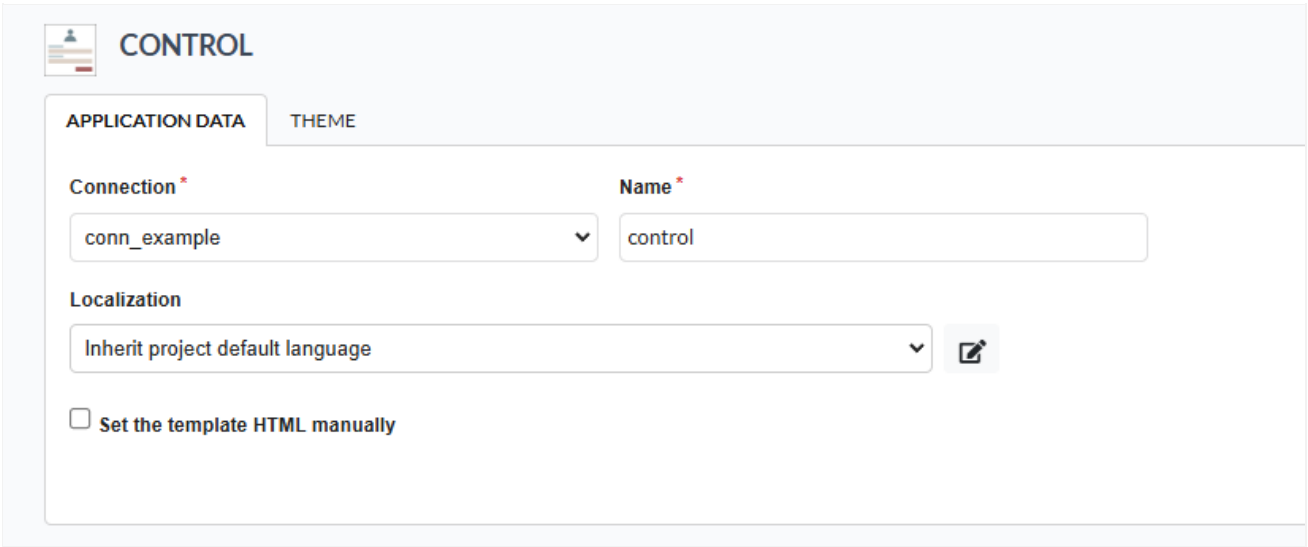
Creating a Control Application

New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.

Application Data



Connection

Defines the connection used for the creation of the application. The existing connections in the project will be listed.

Name

The name of the application being created. It cannot contain special characters.

Localization

The language of the application to be created. The project's default language is automatically selected.

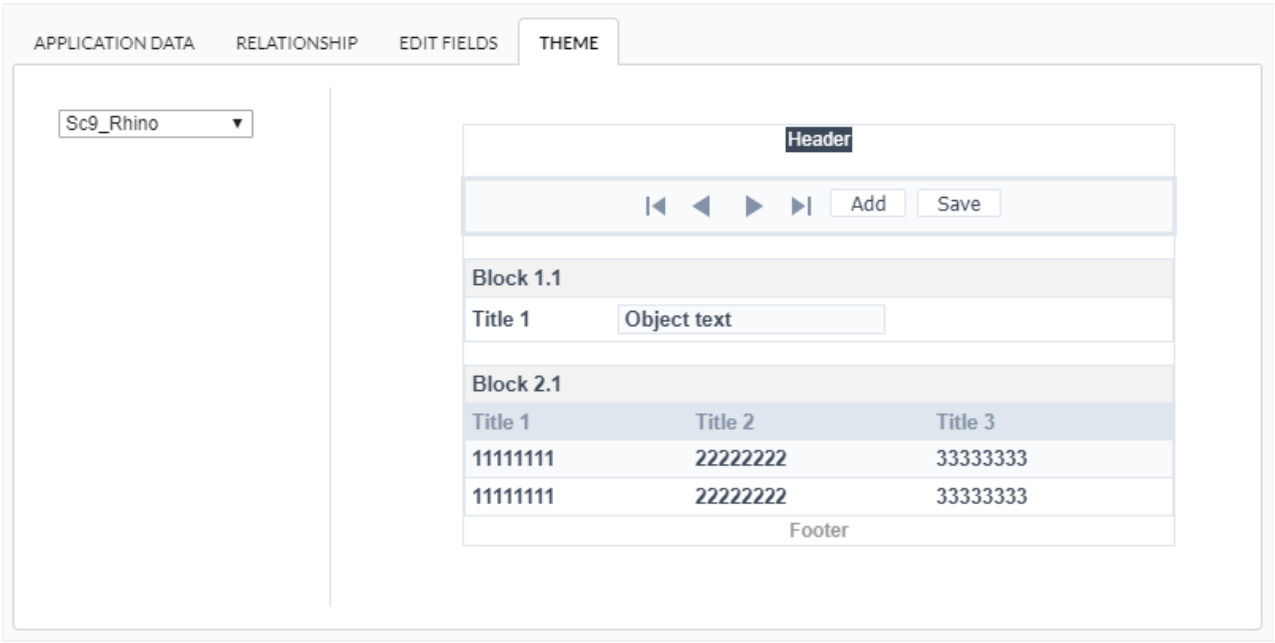
Set the template HTML manually

This option allows you to fully customize the HTML design of the control application, replacing the default template generated by Scriptcase.

The developer must correctly reference the fields and buttons in the customized HTML using placeholders to ensure proper functionality. Knowledge of HTML and CSS is required to create and adjust the design.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.



Related Links

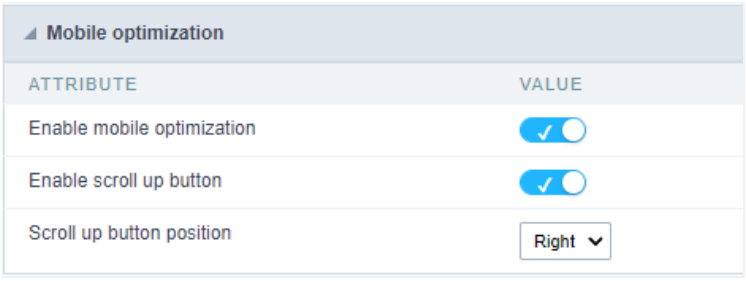
Related Videos

Control Mobile Optimization

The Mobile settings allows to define the automatic optimization of generated applications to run on mobile devices.

See below the available settings.

Enable mobile optimization



This option changes how the application HTML elements works, adapting them automatically to run on mobile devices.

It enables this option by default when creating a new application.

See some examples of adapted screen:

Enable back to top button

It enables the displaying of the button **back to top button** in the Grid, Filter, Detail and Summary modules.

It enables this option by default when creating a new application.

Back to top button position

Defines the position of the **back to the top** button:

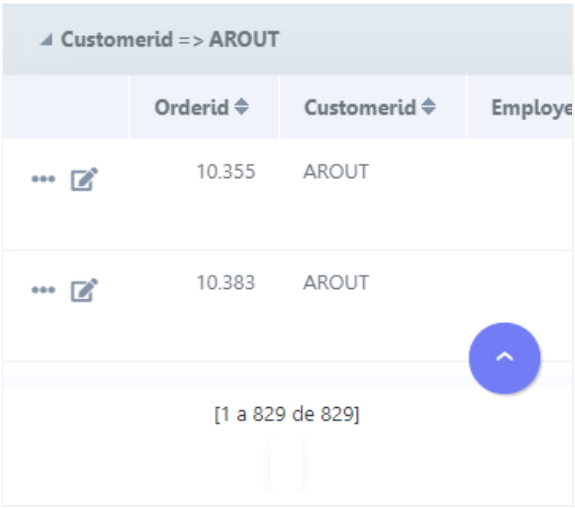
The options are:

- **Right** - Position the button in the bottom right corner;
- **Left** - Position the button in the bottom left corner;

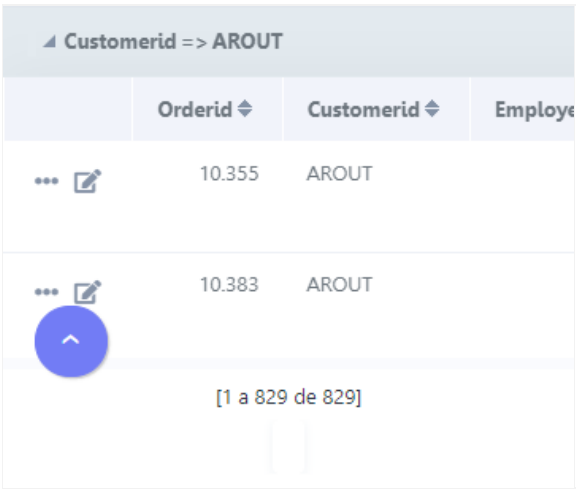
When creating a new application, the button is configured on the right by default.

See below for placement examples.

Button in the bottom right corner



Button in the bottom left corner

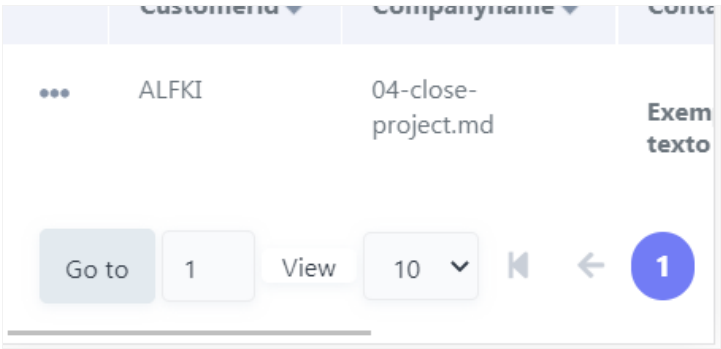


Pin the bottom toolbar

Defines the behavior of the lower toolbar when running the application on mobile devices with the **Simplified toolbar** option disabled.

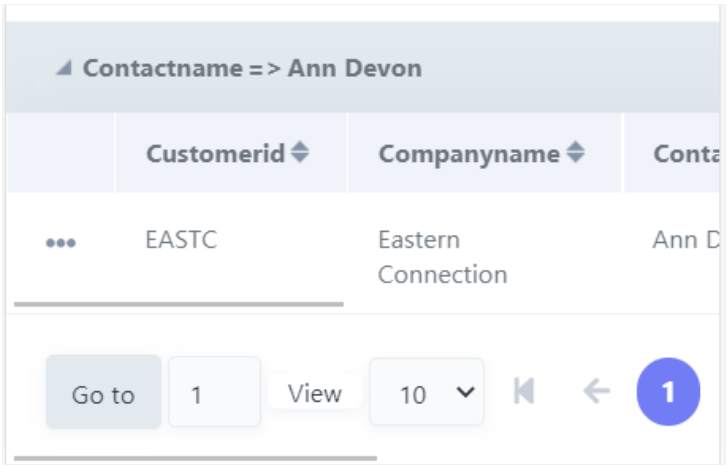
When enabling these options, the lower toolbar will behave similarly to the upper toolbar, remaining fixed on the screen and thus facilitating access to the configured resources.

Example of fixed bottom toolbar



If you choose to uncheck this feature, the lower toolbar will be fixed at the end of the application, so access to the configured buttons will only be available when scrolling to the end of the page.

Example of Traditional Toolbar



This setting is available when the **Simplified Toolbar** options is turned off.

onExecute - Grid

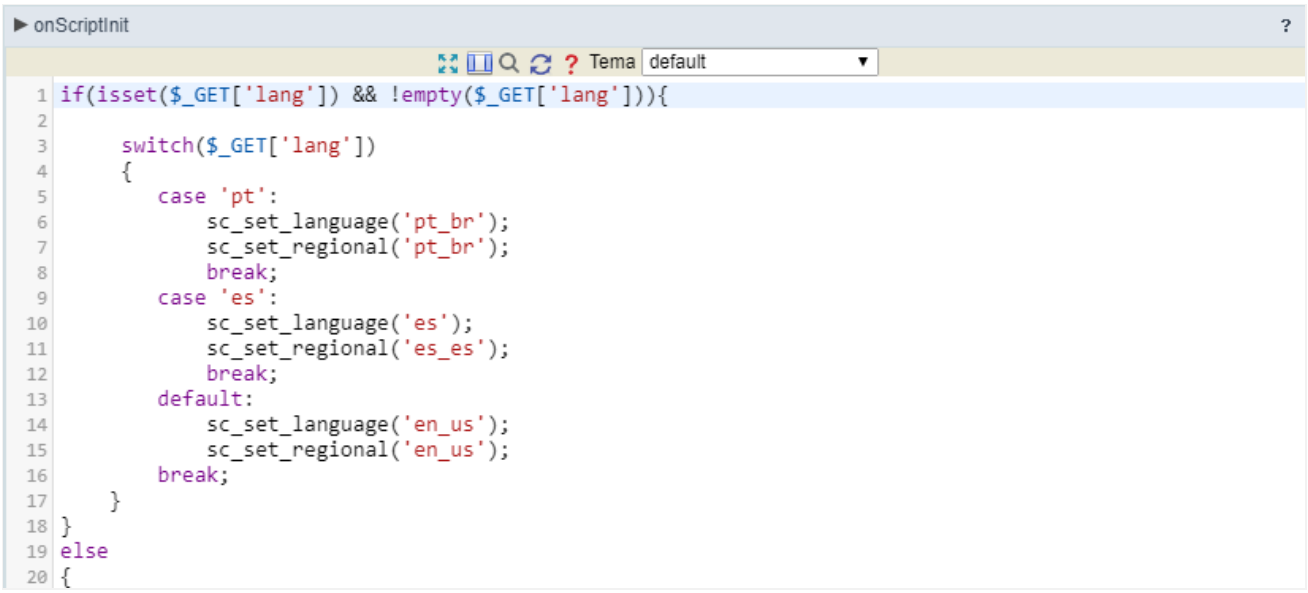
OnExecute

This is the only event for the Blank application, and it occurs every time it runs. It is used to create all the HTML, PHP and JS code for the page.

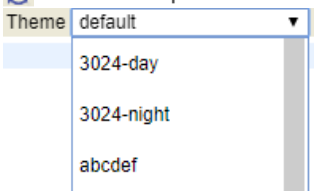
Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.



At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
- 

- Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

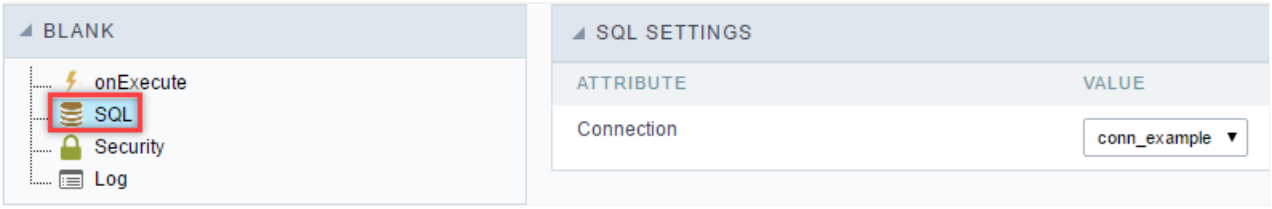
Click Here to view the Scriptcase hotkeys documentation.

Related Links

Related Videos

Blank SQL Settings

In blank applications you can use one connection to access the database by default but as in other application types (Forms, Grids and Control) you can run SQL commands using other connections different from the native application connection.

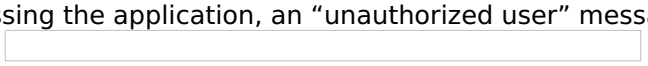


SQL Settings.

- **Connection** : This option allows you to choose one of your project databases connections.

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application. 

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Headers

Disable XSS Auditor

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

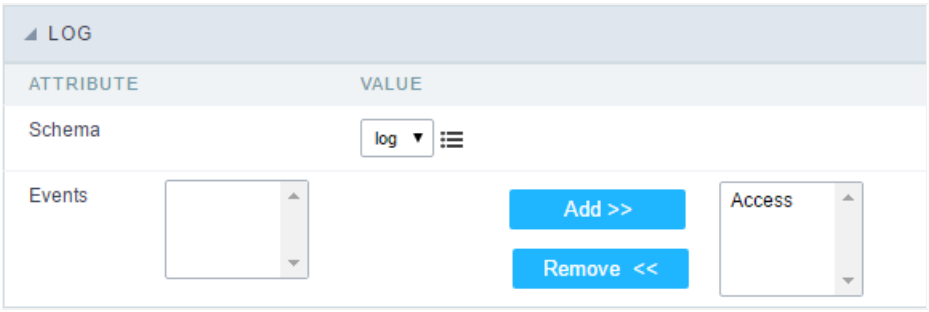
Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Blank Log

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Application Settings (Blank Application)

Settings

Settings

ATTRIBUTE	VALUE
Application Code	blank (9.08.0011)
Application title	{lang_othr_blank_title}
Friendly URL	
Description	
Application images	
Language	Portuguese (Brazil)
Use Shared Localization Variable	✓
Charset	
Share Theme Variable	✓
Folder	root
Edit by Project	✓
Timeout	

Application Code

Informs the current name of the application and the version in which it was created

Application Code	menu (9.06.0014)

Application title

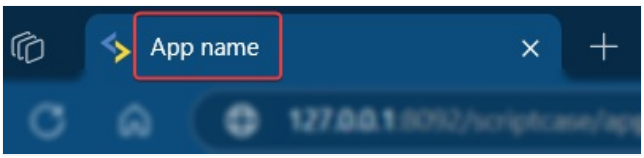
Application Title

Application Title	{lang_app_name}

Defines the application title, which can be set dynamically using langs or with a fixed text.

In this option it is also possible to use langs, for a system with multiple languages. Click here and see how to configure the langs

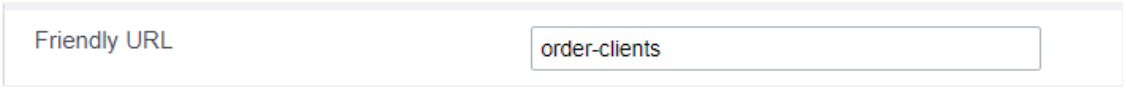
Example of the application title



Friendly URL

Friendly URL

This attribute defines the application's friendly URL.



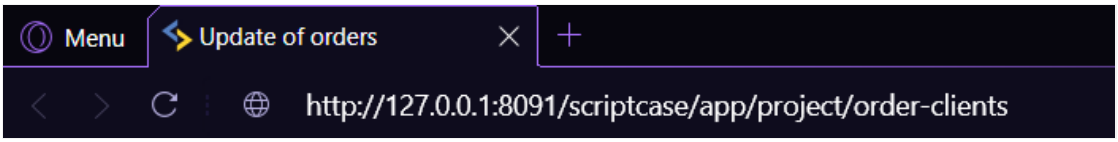
Alphanumeric characters and some special characters are allowed, such as: **hyphen** (-), **underscore** (_), **comma** (,), and **dot** (.).

The use of accents or spaces is not allowed.

Some recommendations for definition are:

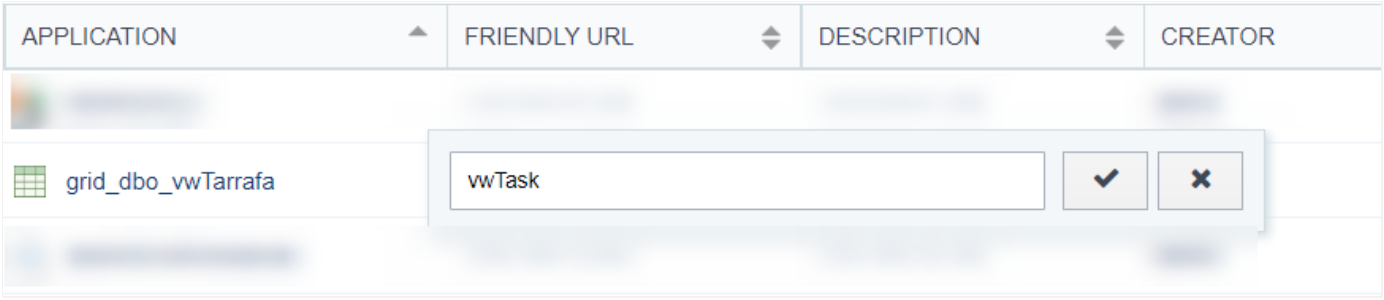
- Use keywords for identification.
- Use hyphens to separate words.
- Use only lowercase letters.
- Avoid using dates.

Example of an application using a Friendly URL



The friendly URL can also be defined in the application list on the project's initial screen.

See the example below



Description

Displays the description of the application.

You can edit or add a description also in the Project Explorer interface.

Application images

It stores the images that will be used in the application through codes in the events, causing these images to be sent along with the application at the time of publication.

Only images used in codes, such as creating a dynamic menu with icons or creating a bill of exchange using a blank, need to be added.

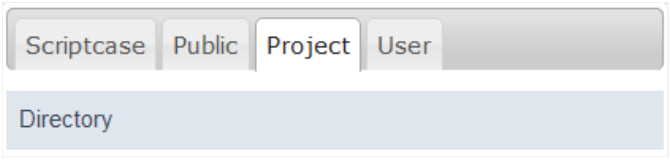
Images added in HTML Image fields or in the application header, for example, are already sent with the applications.

The added images will be stored in this directory: **../_lib/img/**

How to use the images

When inserting the image in the image manager, its name is changed according to the **scope** (public, project or user) and the **storage directory** (background, button, icon, menu or general) following the following pattern.

Scope: It refers to the level of access to the file within Scriptcase, in the development environment.

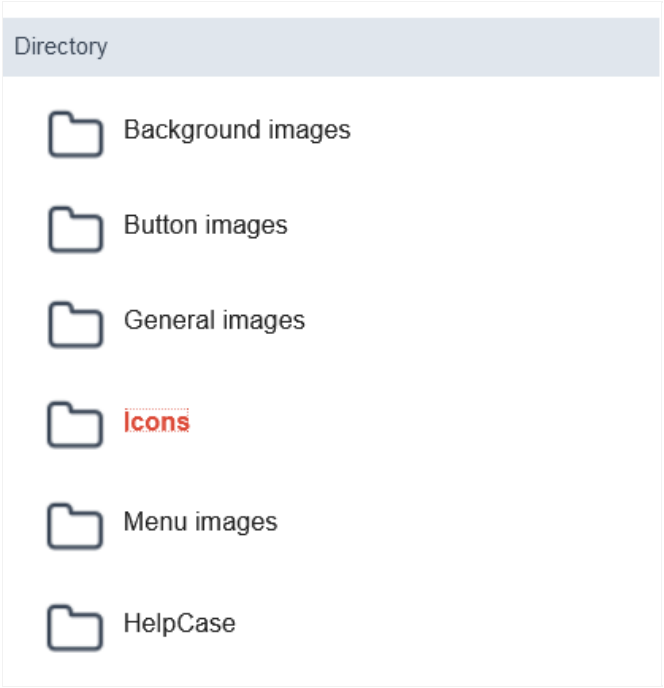


- Scriptcase__ - Contains preexisting images in the tool, where it is not possible to upload new images.
- Public__ - Images accessible for all projects.
- Project__ - Images accessible only in the project in which it was inserted.
- User__ - Images accessible only to the user who uploaded the image.

Each scope adds a prefix to the image name.

- Public - **sys__NM__**
- Project - **grp__NM__**
- User - **usr__NM__**

Storage Directory: Directory where the image will be added. Each directory adds one more prefix to the image name.



- background images - **bg__NM__**
- button images - **btn__NM__**
- general images - **img__NM__**
- icons - **ico__NM__**
- menu images - **menu_img__NM__**

This way, an image called _001.jpg, added in the public scope and in the icon directory will be named **sys__NM__ico__NM__001.jpg**

Language

Defines a specific language for the application, overriding the default language defined in the project. If no language is selected, the application will inherit the project's default language.

The languages listed in the option are defined in the project properties

Use Shared Localization Variable

Defines if the language of the applications will be stored

Charset

Defines an application-specific charset.

If no value is selected, the application inherits the project's default charset.

Ideally, this charset option, within the application's settings, does not have a defined value. Different usage of your base charset can cause problems saving information in your tables.

Share Theme Variable

When active, this option allows the theme stored in the session to be used by the application.

For example, by selecting theme x at login, all applications in the project will use the selected theme.

If the option is unchecked, the application remains with its original theme.

Folder

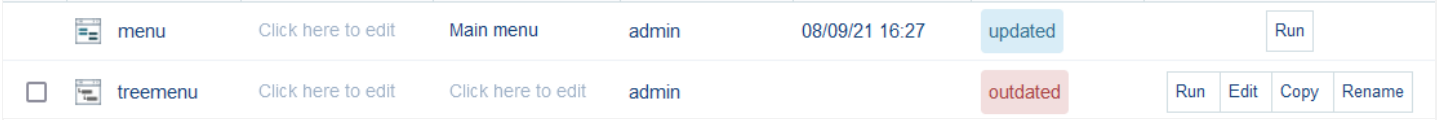
Folder where the application is stored in Project Explorer.

Editing by Project

By disabling this option, only the user who created the application will have access to its settings.

No other project users will be able to access the application.

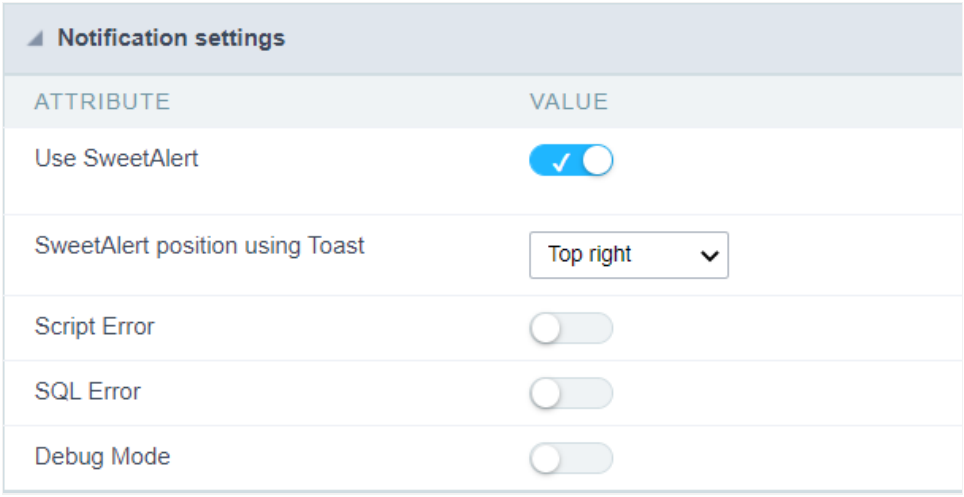
In the image below, the menu application has the edit by project option disabled.



Timeout

Set the session runtime timeout in seconds. If the value is Zero, it assumes the default timeout of the PHP.

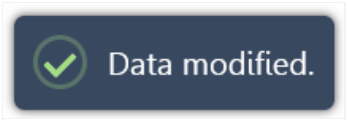
Notifications Settings



Use SweetAlert

Activate SweetAlert in the application, replacing the “confirm” and “alert” of the browser.

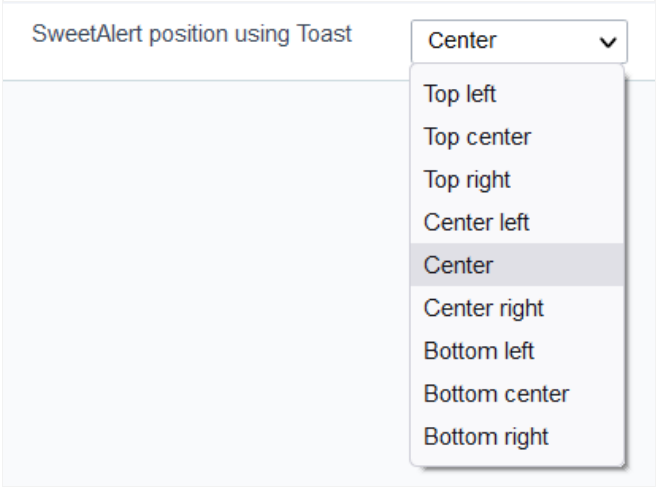
Confirm Example with SweetAlert Enabled



Confirm Example with SweetAlert Disabled

SweetAlert position using Toast

Sets the display position of the message window using toast.



Script Error

Allows displaying the line code where there is an error.

SQL Error

Allows displaying the SQL statement if it got an error

Debug Mode

Runs the application in Debug mode, showing all SQL statements the application is executing.

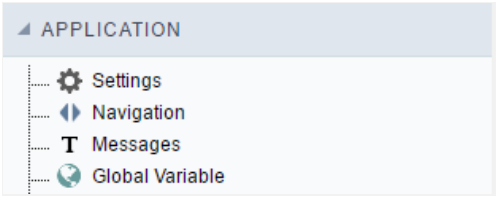
Global Variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target="_blank"} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



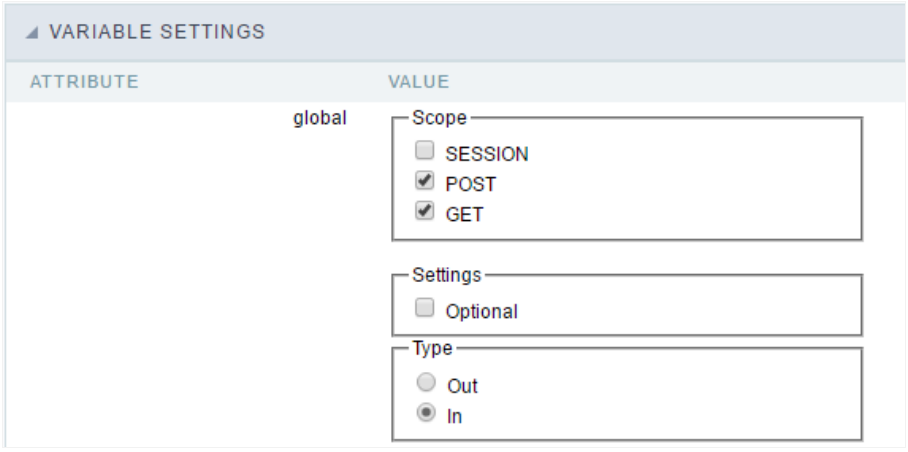
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

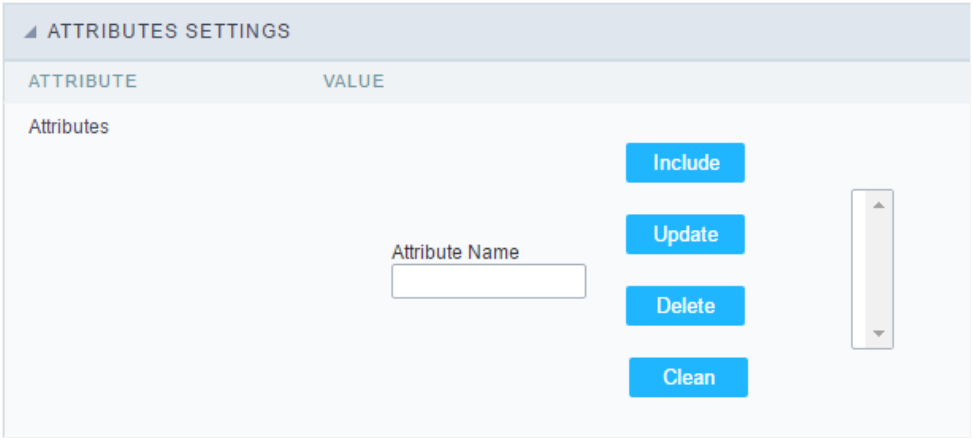
Indicates where the application is using the variable.

Blank Programming

ScriptCase has incorporated the concept of Object Oriented programming, using attributes, resources, methods and libraries. It is possible to create your own business rules in applications, and by using these concepts you can reap huge rewards in terms of better organization and improved development.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

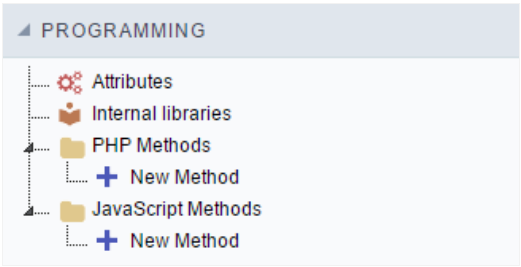
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

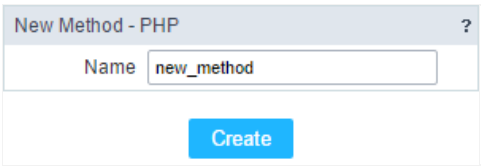
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

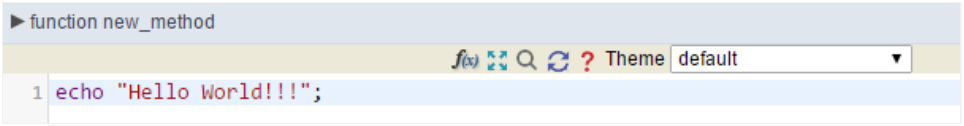


- Give the method a name and click Create. As in the image below.

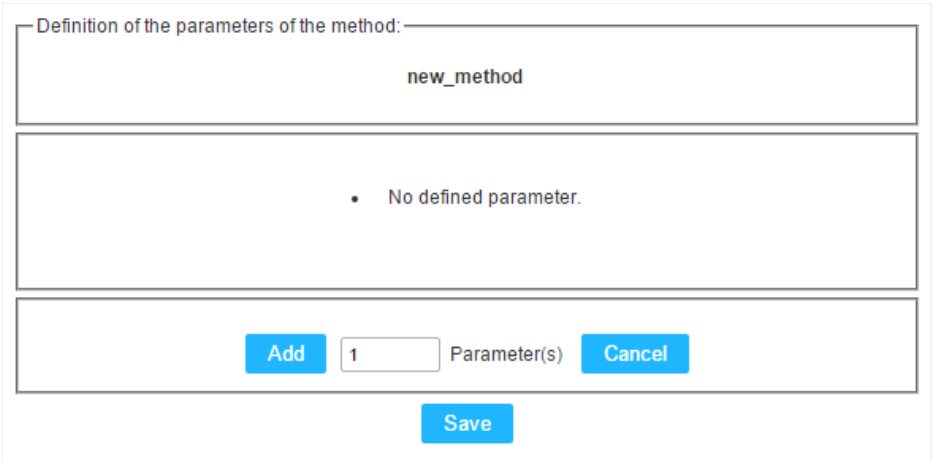
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



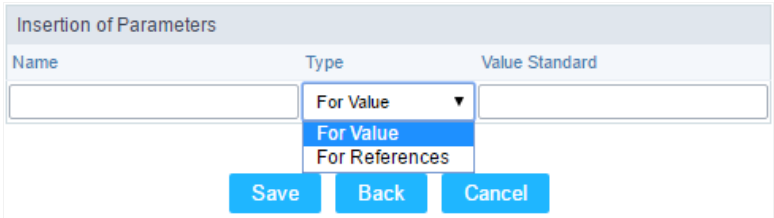
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

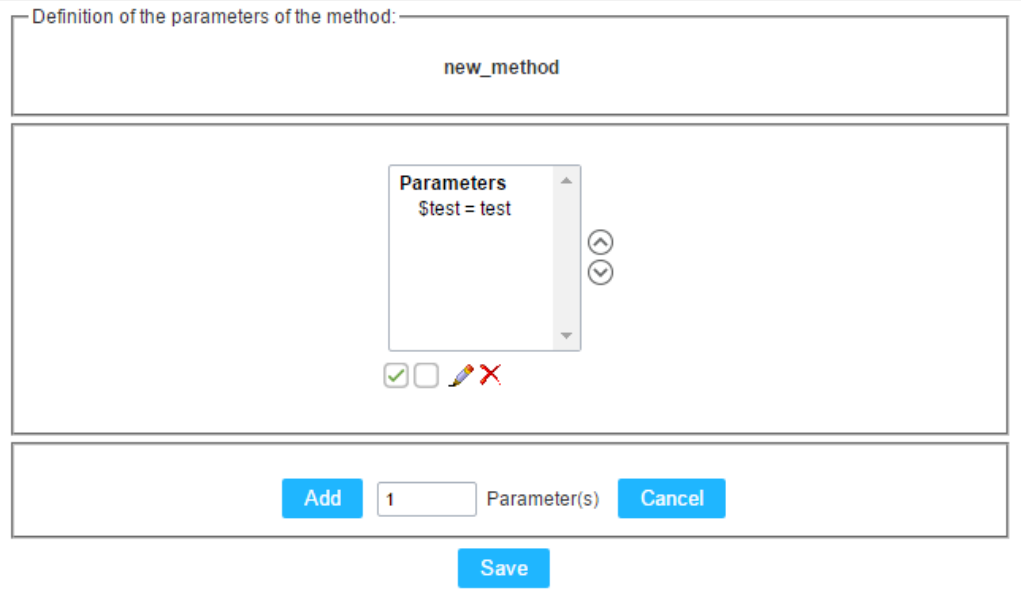


- **Name** : Type in the variable's name.
- **Type** : Selecting the type of variables: For Value or For Reference.

- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:



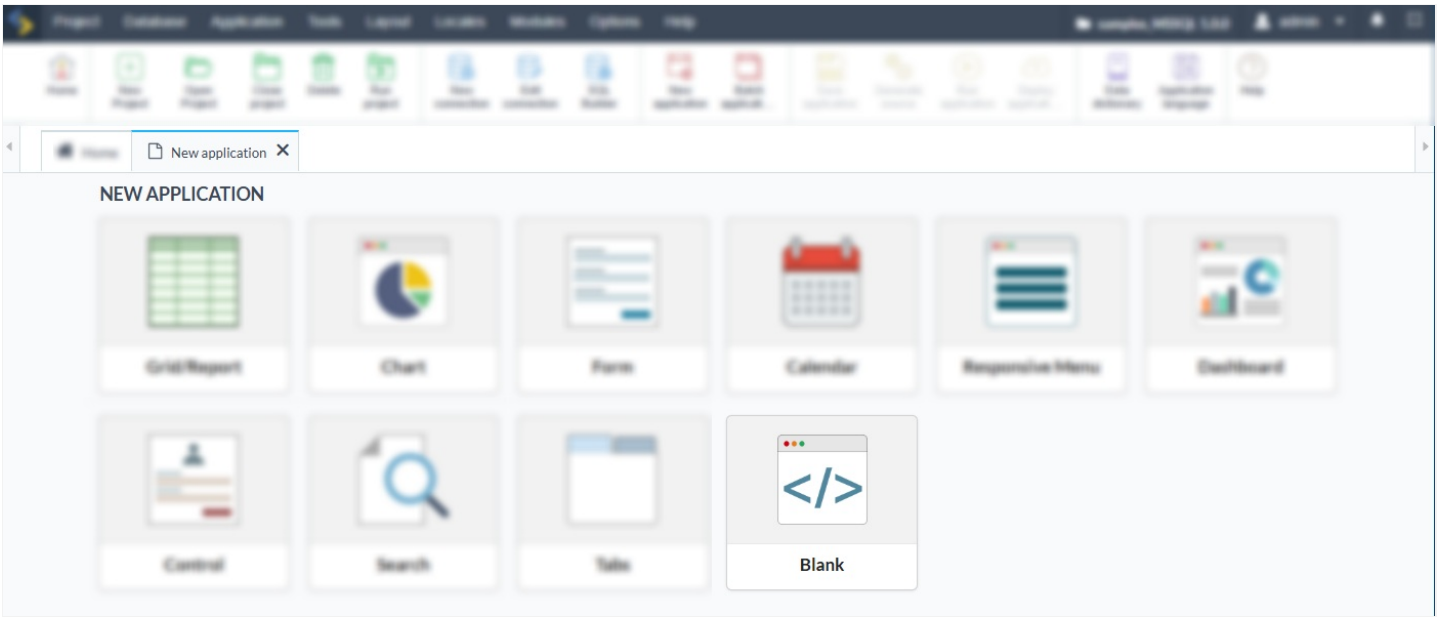
- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

Creating a Blank Application

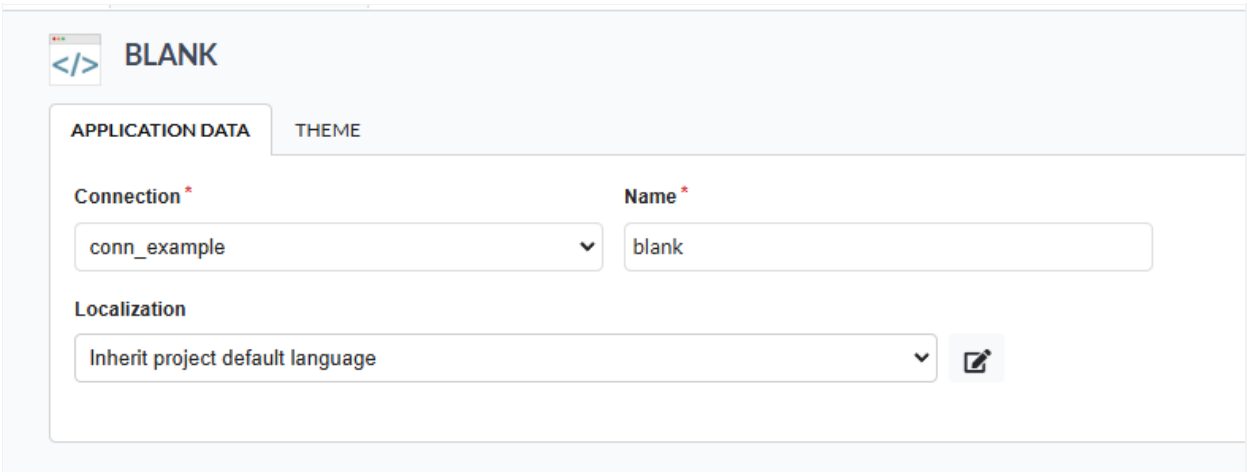
New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.



Application Data



Connection

Defines the connection used for the creation of the application. The existing connections in the project will be listed.

Name

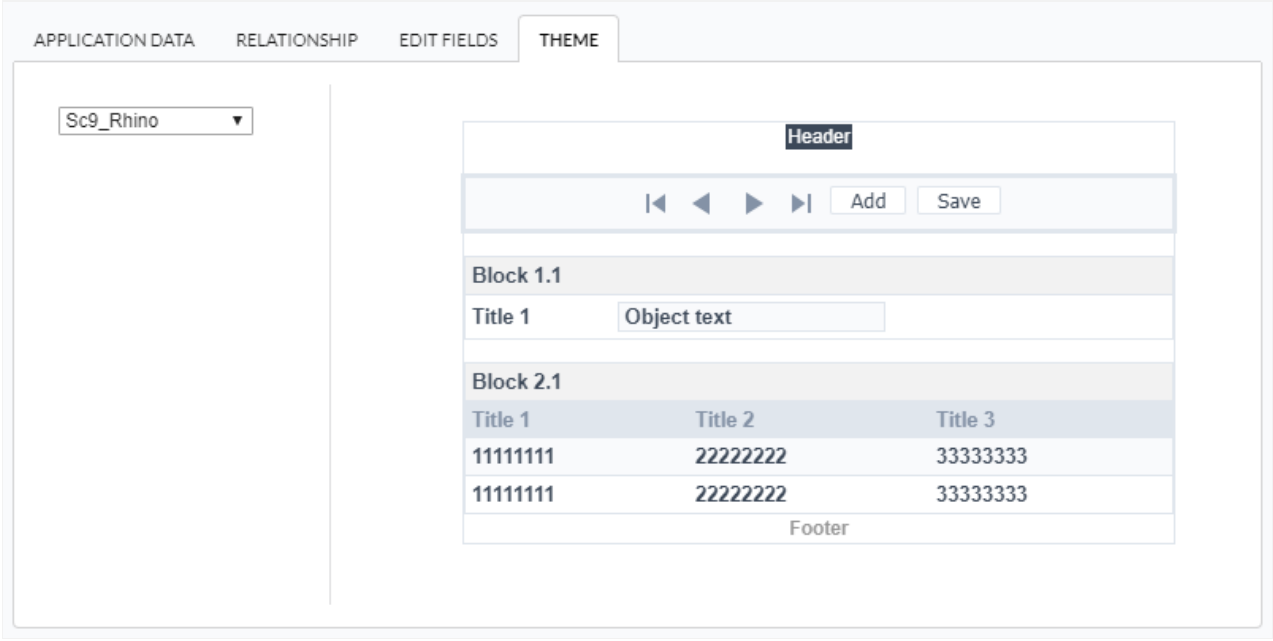
The name of the application being created. It cannot contain special characters.

Localization

The language of the application to be created. The project's default language is automatically selected.

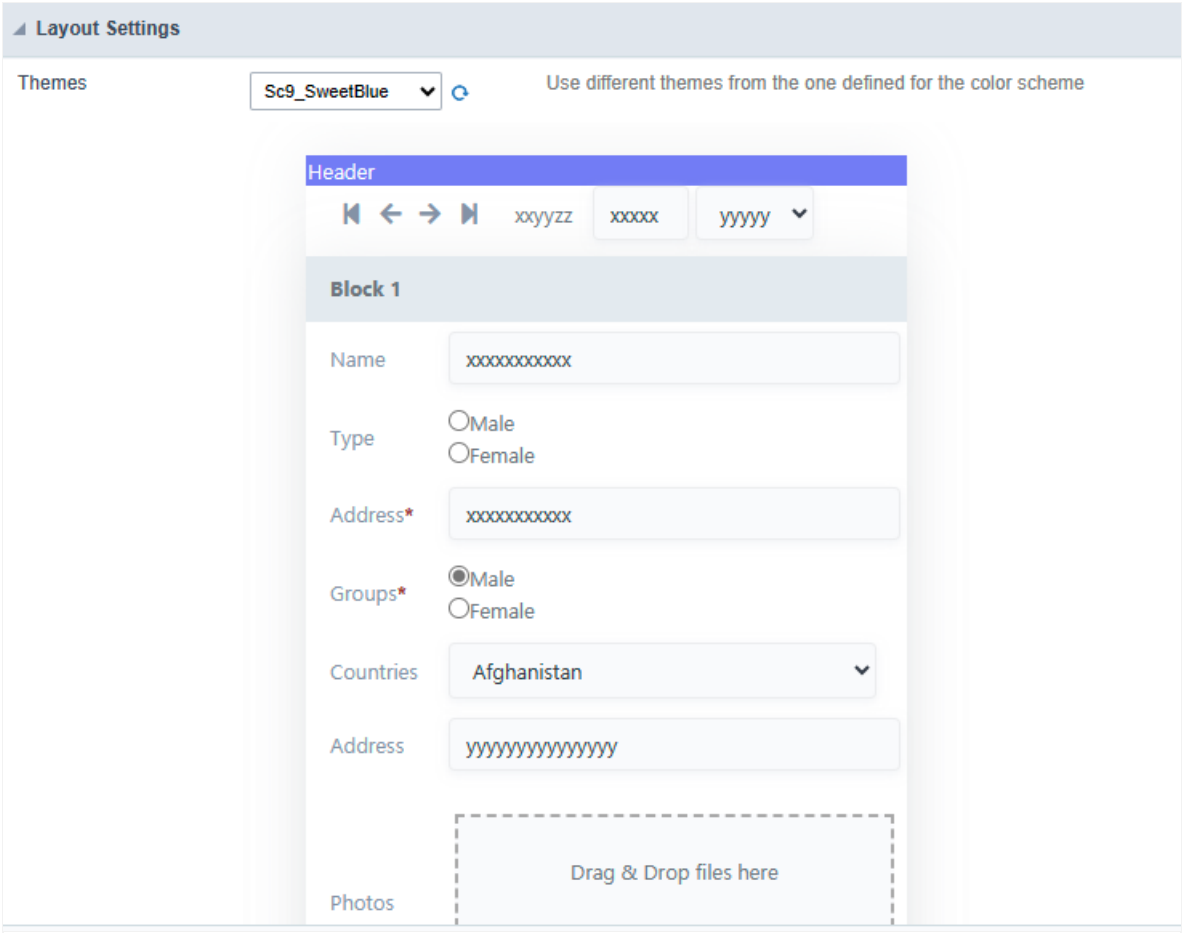
Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.



Blank Layout Settings

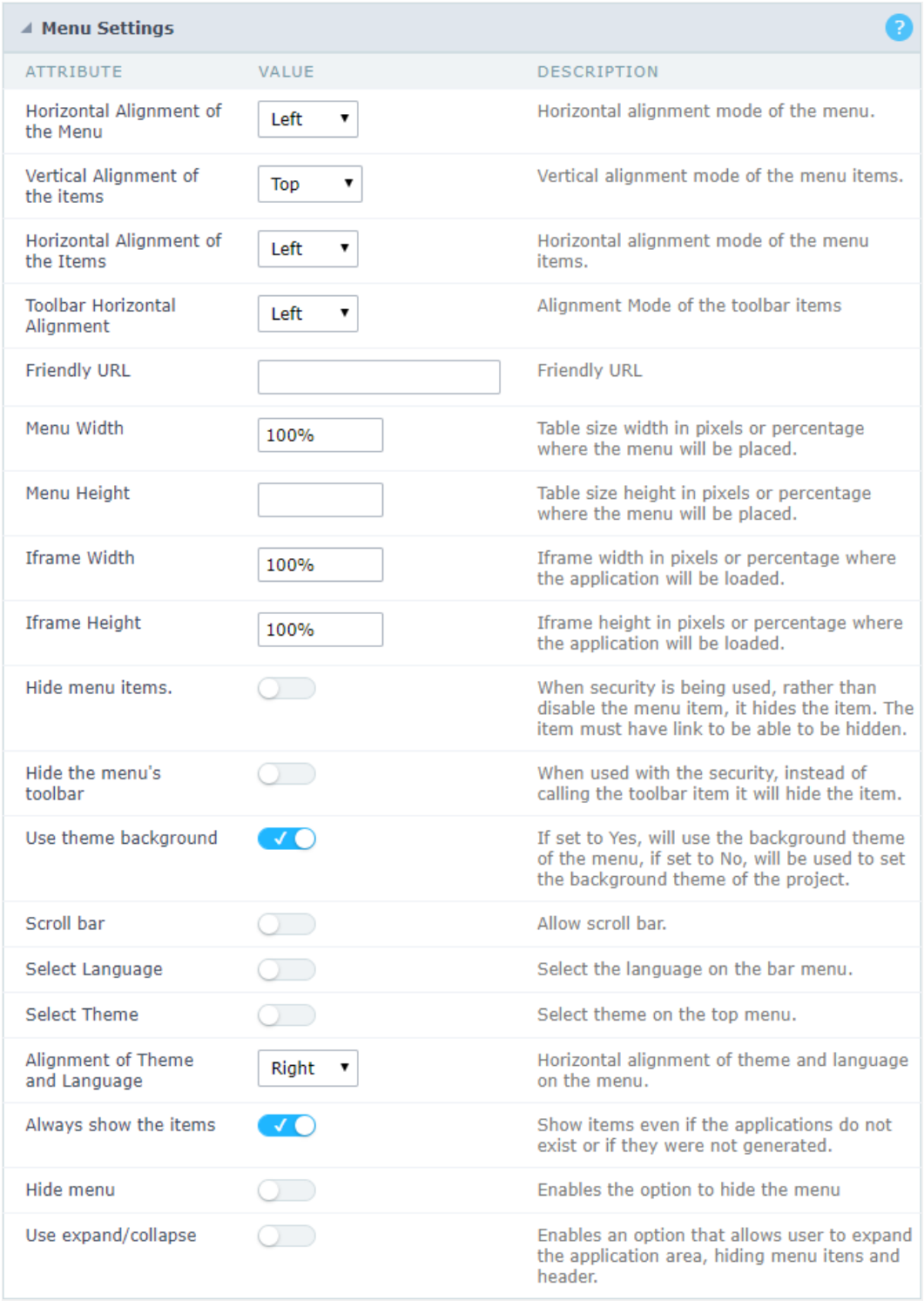
Defines the theme used to display messages in the application.



Menu Settings

Desktop

The menu settings allow to define the attributes of the application display, according to the following:



Menu Settings

ATTRIBUTE	VALUE	DESCRIPTION
Horizontal Alignment of the Menu	Left	Horizontal alignment mode of the menu.
Vertical Alignment of the items	Top	Vertical alignment mode of the menu items.
Horizontal Alignment of the Items	Left	Horizontal alignment mode of the menu items.
Toolbar Horizontal Alignment	Left	Alignment Mode of the toolbar items
Friendly URL		Friendly URL
Menu Width	100%	Table size width in pixels or percentage where the menu will be placed.
Menu Height		Table size height in pixels or percentage where the menu will be placed.
Iframe Width	100%	Iframe width in pixels or percentage where the application will be loaded.
Iframe Height	100%	Iframe height in pixels or percentage where the application will be loaded.
Hide menu items.		When security is being used, rather than disable the menu item, it hides the item. The item must have link to be able to be hidden.
Hide the menu's toolbar		When used with the security, instead of calling the toolbar item it will hide the item.
Use theme background	✓	If set to Yes, will use the background theme of the menu, if set to No, will be used to set the background theme of the project.
Scroll bar		Allow scroll bar.
Select Language		Select the language on the bar menu.
Select Theme		Select theme on the top menu.
Alignment of Theme and Language	Right	Horizontal alignment of theme and language on the menu.
Always show the items	✓	Show items even if the applications do not exist or if they were not generated.
Hide menu		Enables the option to hide the menu
Use expand/collapse		Enables an option that allows user to expand the application area, hiding menu itens and header.

Menu configuration Interface.

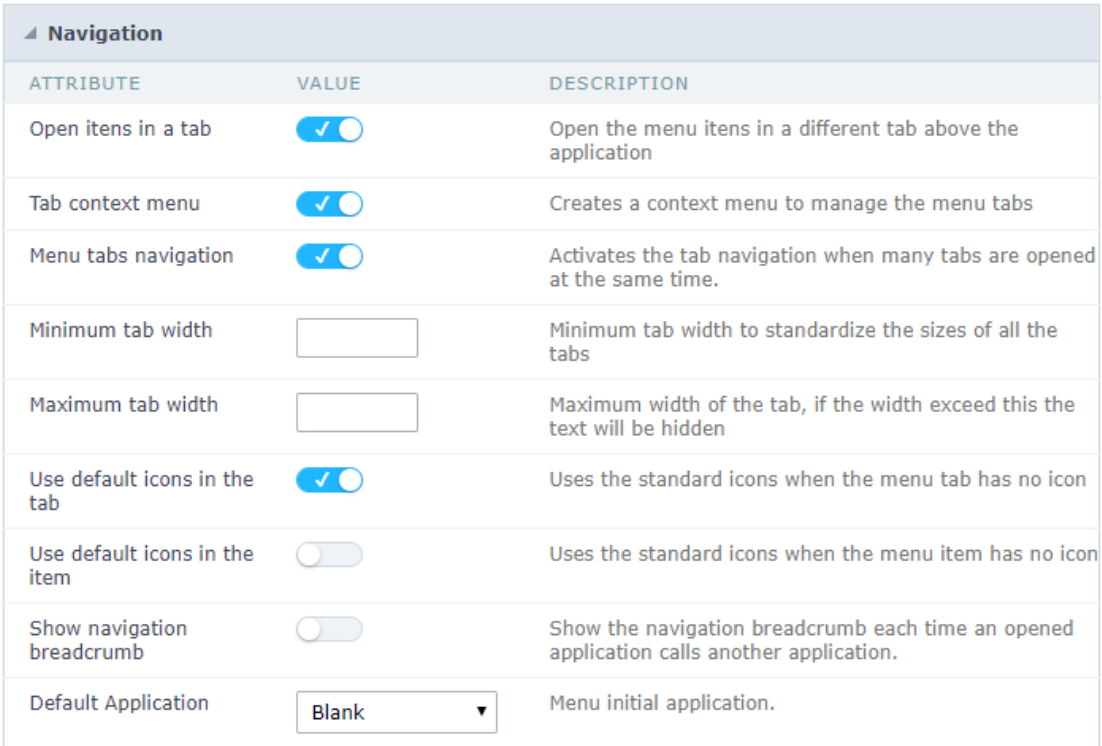
- **Horizontal Alignment of the Menu**: Defines the horizontal alignment of the menu (Left, Center, Right).
- **Vertical Alignment of the items**: Defines the vertical alignment of the menu items (Top, Center, Bottom).
- **Horizontal Alignment of the Items**: Defines the horizontal alignment of the menu items (Left, Center, Right).
- **Toolbar Horizontal Alignment**: Defines the horizontal alignment of the toolbar items (Left, Center, Right).
- **Friendly URL**: Allows defining a name for the URL that's going to be called by the application. The characters allowed in the URL are(a-z, A-Z, 0-9). You still can inform the Friendly URL on the project home, where there is a list of applications with

the “Friendly URL” column.

- **Menu Width**: Define the width of the menu in pixels or percentages.
- **Menu Height**: Define the height of the menu in pixels or percentages.
- **Iframe Width**: Define the width of the Iframe (Where the applications will open in) in pixels or percentages.
- **Iframe Height**: Define the height of the Iframe (Where the applications will open in) in pixels or percentages.
- **Hide Menu Items**: This attribute defines whether the menu item will be displayed or not. When enabled, items are hidden if they do not have a link, the application has active security (without user access), or the item is disabled using the `sc_menu_disable` macro. When disabled, the items will always be displayed.
- **Hide the menu’s toolbar**: When enabled and used with the security module, it’ll hide the toolbar that the user doesn’t have permission to access, instead of only disabling them.
- **Use theme background**: If enabled, it’ll use the background theme of the menu. Else it’ll use the background defined in the project.
- **Scroll bar**: Enables the scroll bar in the Iframe.
- **Select Language**: Allows to select the language in the menu bar.
- **Select Theme**: Allows to select the theme in the menu bar.
- **Alignment of Theme and Language**: Defines the horizontal alignment of the language and theme comboboxes.
- **Always show the items**: Displays the application items, even though they don’t exist, or haven’t been generated yet.
- **Hide menu**: Enables the option to hide the menu.

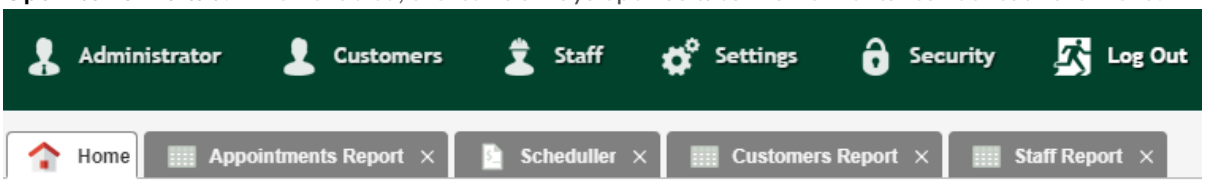
Navigation

The menu navigation settings allow to define the attributes of the tabs and the navigation path, according to the following:



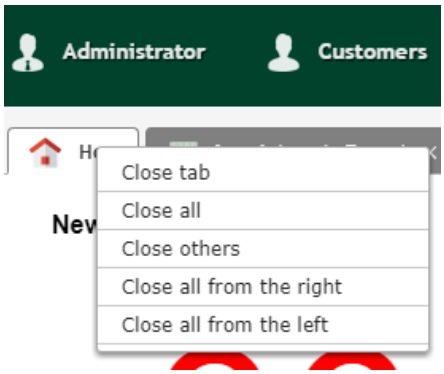
Menu Navigation configurations Interface.

- **Open items in a tab**: When enabled, the items always open as tabs in a horizontal bar beneath the menu.



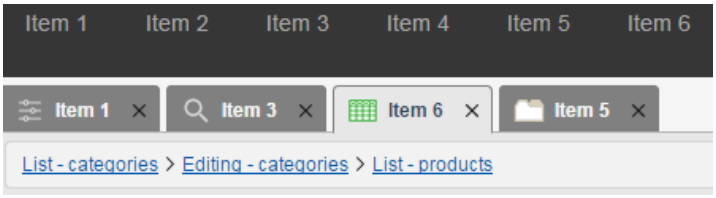
Navigating through Applications with tabs.

- **Context menu of the Tabs**: Allows you to manage the open tabs.



Application with a context menu.

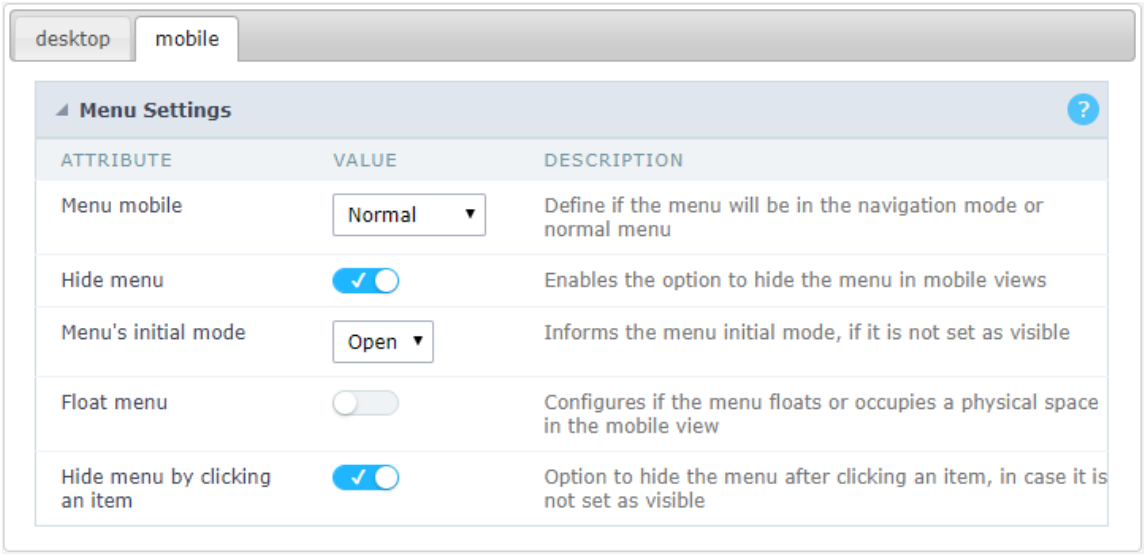
- **Minimum tab-width**: Allows to inform the min-width of all the tabs.
- **Maximum tab-width**: Allows to inform the max-width of all the tabs.
- **Show navigation breadcrumb**: Displays the path containing the application in the tab.



Navigation path Display in the Application.

- **Use default icons in the item**: When enabled, this option displays the standard icons for each item.
- **Use default icons in the tab**: When enabled, this option displays the standard icons for each tab.
- **Default Application**: Select a default application initiate when the menu opens.
- **Show the initial app as a tab**: Allows to open the default application in a tab.

Mobile



Mobile menu configuration Interface.

- **Hide menu**: Enables the option to hide the menu when viewing from a mobile device.
- **Menu's initial mode**: Inform the starter mode of the menu. Otherwise, it'll always be visible.
- **Hide menu by clicking on an item**: When enabled, it hides the menu when clicking on an item.
- **Float Menu**: Hides the icon to open the menu or always displays it.

Related Links 🔗

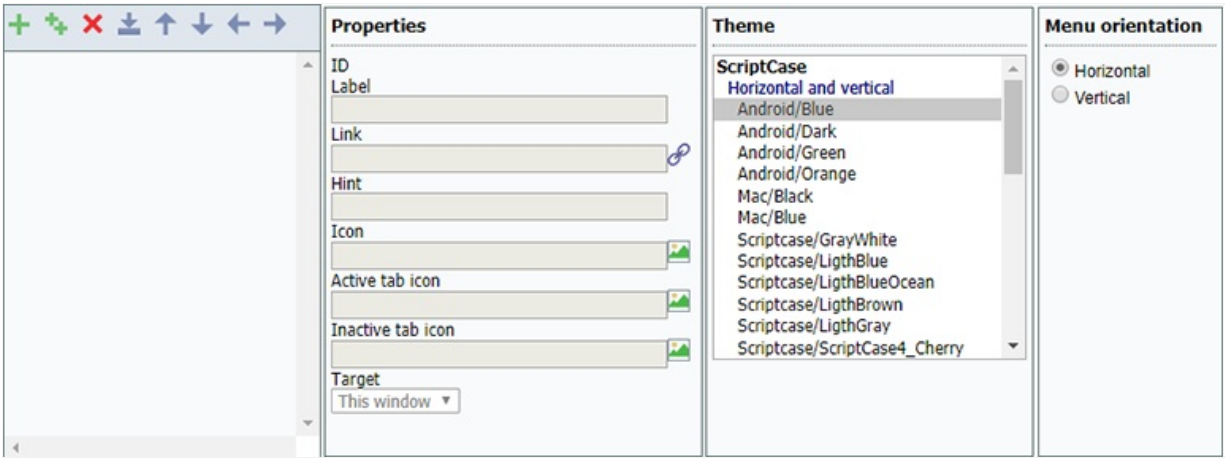
Related Videos ▷

Menu items

Menu items settings

In this screen the user menu structure, indicating which applications will be called. Presents the following attributes, which should be completed by the user:

On mobile devices, if the menu item has sub-items and link, the link will be scorned to display the sub-items.



Menu settings interface.

- **Label** : Title for the application within the menu. Type a fixed title or use the Scriptcase language system to create lang variable for these names, according to the languages used in the project.
- **Link** : Here you have to enter the application name that will be linked to the menu item. You can also click the search button and select from the application list. This link can also be an external URL.
- **Hint** : Descriptive text that helps the user to identify the application. It is an optional field.
- **Icon** : Image that will be displayed as an icon in the menu item. The user can click the search button to select the image from the Scriptcase Image Manage. It is an optional field.
- **Active tab icon** : Image that will be displayed as an icon in the menu item when active. The user can click the search button to select the image from the Scriptcase Image Manage. It is an optional field. If you leave it blank Scriptcase adds a default image.
- **Inactive tab icon** : Image that will be displayed as an icon in the menu item when inactive. The user can click the search button to select the image from the Scriptcase Image Manage. It is an optional field. If you leave it blank Scriptcase adds a default image.
- **Target** : This option sets how the application will be displayed, it can be targeted as: Same window, Another window or leave (it will log the user out from the system).

Resources for menu items creation

- **Insert item** : This option add items to the menu.
- **Insert Sub-Item** : This option add sub-items to the menu, you might select the main item before to add the sub-items within the selected main item. You can also manage the sub-item position using the arrows.
- **Remove** : This option deletes the item/sub-item, you can select the items before and then remove.
- **Import applications** : You can use this option to call an application to attach to the item. It will open the project application list.
- **Move up** : This option allows you to move up a menu item or sub-item.
- **Move down** : This option allows you to move down a menu item or sub-item.
- **Move left** : This option allows you to move a menu item or sub-item to the left. You can use it to organize the menu hierarchy.
- **Move right** : This option allows you to move a menu item or sub-item to the right. You can use it to organize the menu hierarchy.

Importing the Applications

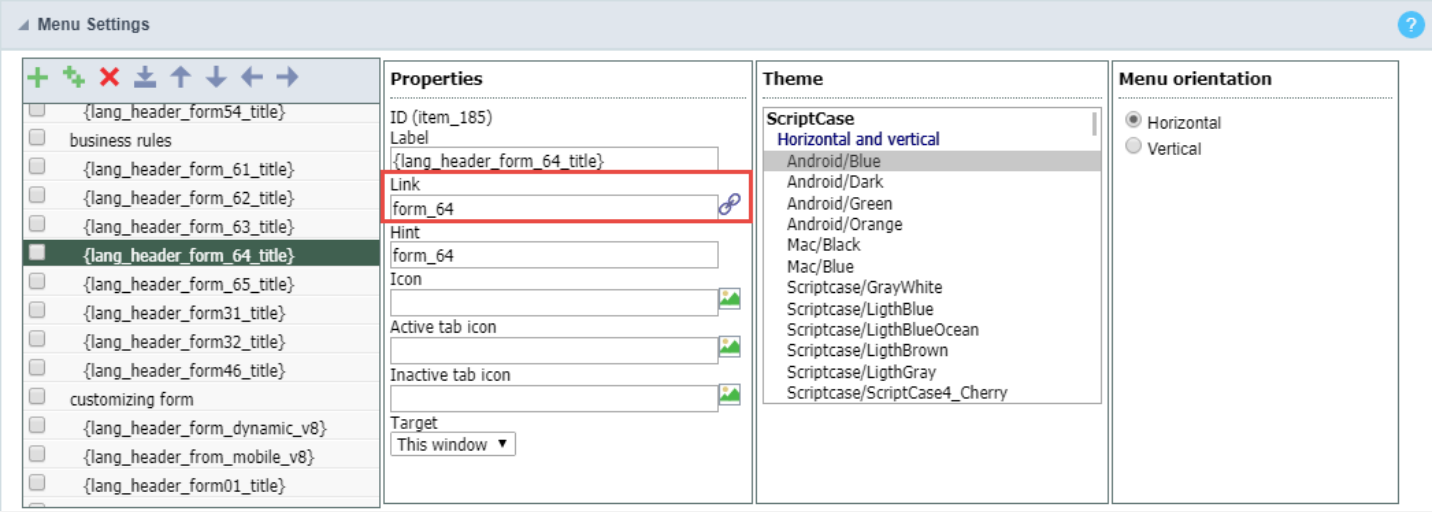
Importing applications is easy and fast. You can select to import one or more applications from a list. Then you need to organize the menu levels as you wish, by using the positioning buttons.

samples

Application	Directory	Dt. Creating	Description
calendar01	root/Calendar	14/07/10 17:15	Calendar Application *
calendar02	root/Calendar	19/07/10 18:43	
calendar03	root/Calendar	29/10/10 17:32	Calendar per user
calendar_color_event	root/Calendar	25/07/14 00:11	Calendar Application with print button and event color per day *
chart01	root/Grid_Applications/Charts	25/01/13 10:02	Bar HTML 5 Chart *

Applications list to Import.

The link property receives the application name, as the image below.



Application Import Interface.

Related Links

Related Videos

Menu Toolbar

This article describes the menu application toolbar and provides information on how to add new items and how to configure them.

Toolbar buttons

Resources for creating menu items Below are the options for adding items to the toolbar. Added items are displayed in the list of items.

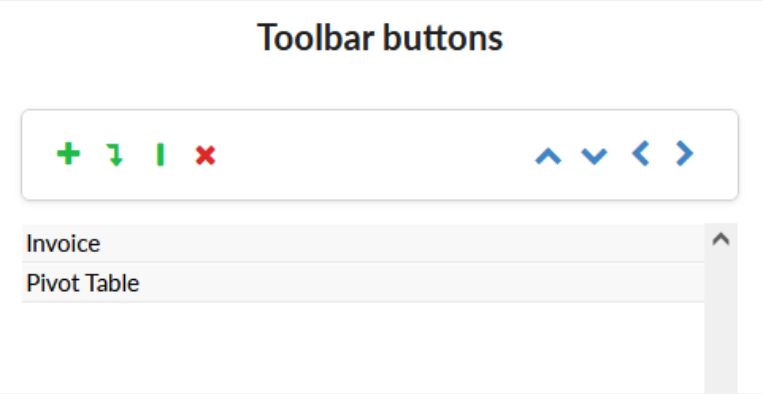
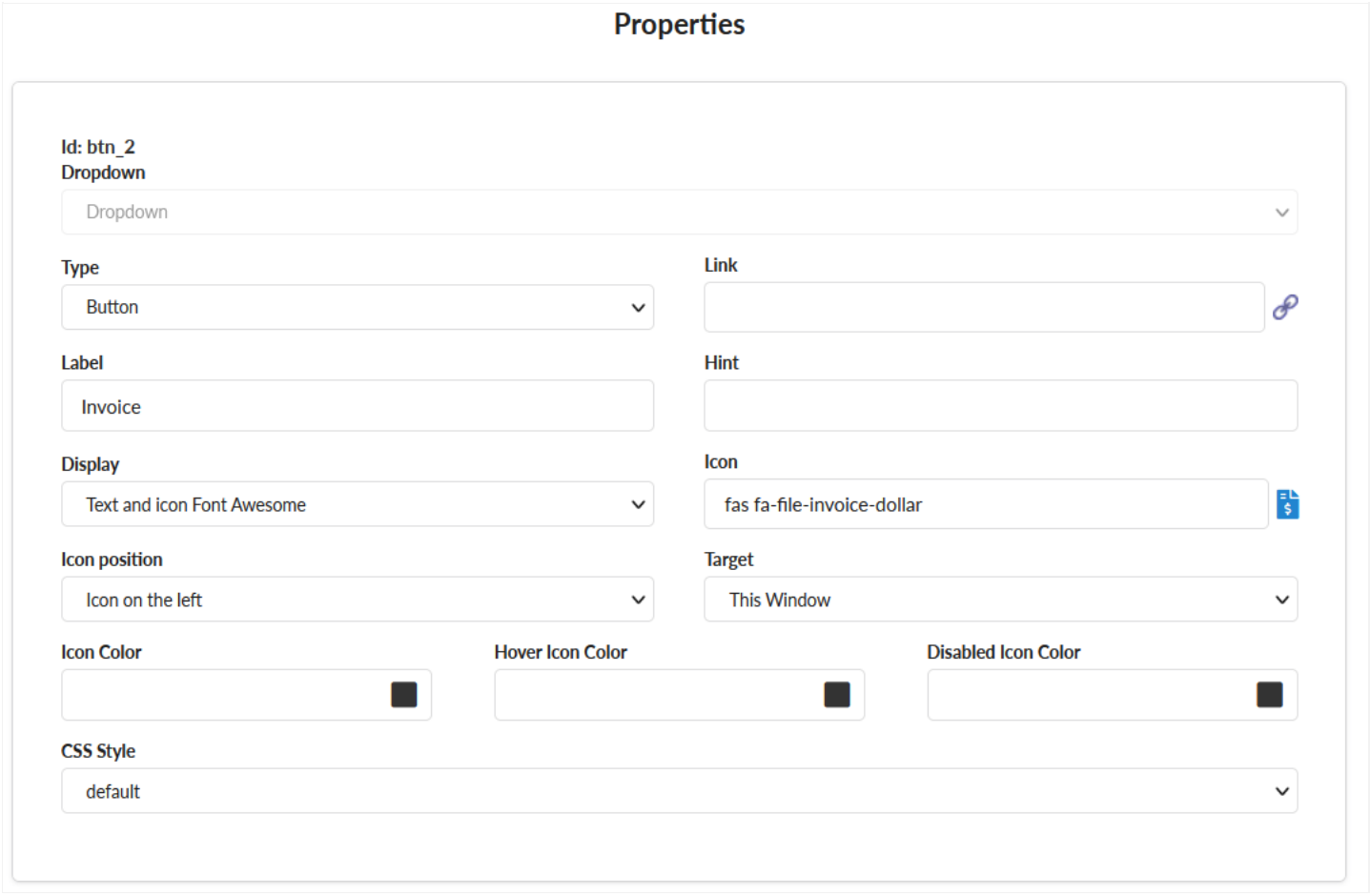


Table with the description of each of the options for creating and ordering the toolbar items.

Title	Icon	Description
Insert item		Add a new item to the menu toolbar.
Insert subitem		Add a sub-item to the menu toolbar. For this, some item must be previously selected. Otherwise an item will be added.
Add Separator		Adds an item separator to the toolbar.
Remove		Remove the selected item from the toolbar
Move up		The move up button changes the order of the selected item, moving it up.
Move down		The move down button changes the order of the selected item, moving it down.
Move left		The move left button changes the position of an item, moving it to the left. In this way, an item added as a sub-item can be turned into an item.
Move right		The move left button changes the position of an item, moving it to the right. In this way, an item can be turned into a sub-item. This button is only active if the selected item is positioned below another item In the toolbar it is only possible to create one level of items.

Button Properties

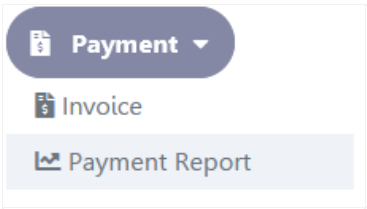
Properties are available by selecting one of the previously added toolbar buttons.



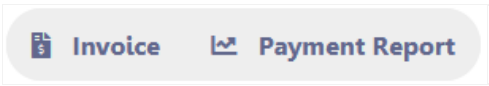
Dropdown

Defines the display format of subitems

Dropdown - The subitems displayed in the list.



Side by side - Subitems will be displayed side by side in the toolbar.



Type

Defines the type of button that will be displayed.

The available options are:

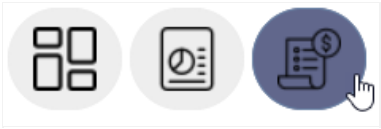
Image

When selecting this option only the image will be displayed, without any CSS application to the button.

Below is an example of a button configured as an image

Button

Below is an example of a standard button



Link

In this option, only the button's label will be displayed in the form of a link, without any application of css to the button.

Below is an example of a button configured as a link

Link

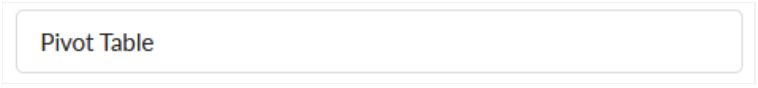
Allows selecting the application through his name or by a link.

Label

Defines the title that will be shown on the button when the application is run.

In addition to a fixed text, the **Label** attribute allows the use of langs to define the button's title, making it possible to use your application in a multi-language project.

Using a fixed text



Using a lang

Click here for more details on Application Language

Hint

Displays a message in the hover of the button.

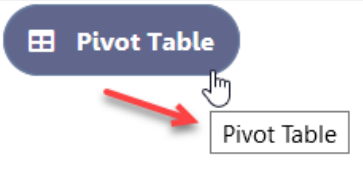
For the feature to work, just add a text or language in the **Hint** attribute, as in the example below.

Using a fixed text

Using a lang

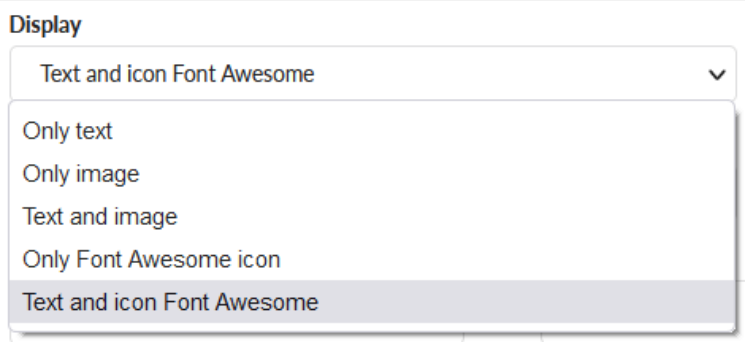
Click here for more details on Application Language

See below the behavior of each of them:



Display

Defines the way the button will be displayed on the toolbar. The options are **Only text**, **Only image**, **Text and image**, **Only Font Awesome icon** and **Text and icon Font Awesome**



See below the behavior of each of them:

Only text

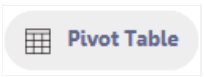
Displays only the text entered in the **Label** attribute.

Only image

Displays only the image informed in the **Icon** attribute.

Text and image

Displays the text and image informed in the attributes **Label** and **Icon**, respectively.

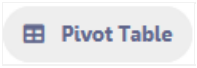


Only Font Awesome icon

Display Font Awesome icon informed in the attribute **Icon**.

Text and icon Font Awesome

Displays Font Awesome texts and icons, informed in the attributes **Label** and **Icon**, respectively.



Icon

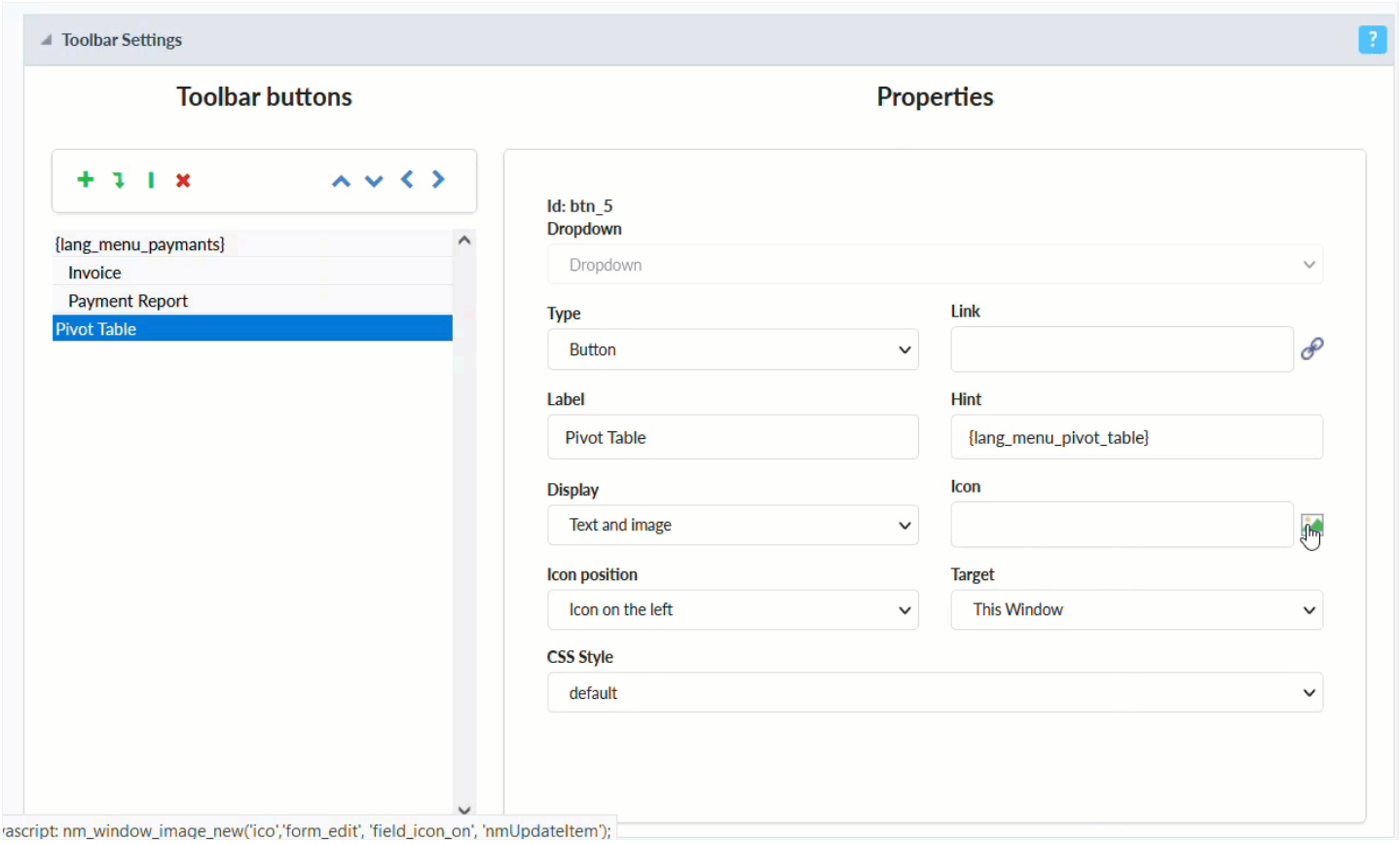
Defines the icon or image that is shown on the button.

This attribute works in two ways, to select a image or Font Awesome icon, it will depend on the option selected in the **Display** attribute.

When selected Only image or Text and image

If on the attribute Display was selected, Only image or Text and image, you need select a image.

See how add a image



When selected Only Font Awesome Icon or Text and icon Font Awesome

If on the attribute Display was selected, Only Font Awesome Icon or Text and icon Font Awesome, you need select a Font

Awesome Icon for the icon attribute.

See how add a Font Awesome Icon

![how add a Font Awesome Icon][gif_selecting_a_font_awesome_icon_attribute]

.png [gif_selecting_a_font_awesome_icon_attribute]:
../../../../assets/images/docs/app/menu/toolbar/selecting_a_font_awesome_icon_attribute.gif .png

Icon Position

Defines the position of the Text or Image (Text to the right, Image to the right).

Target

Allows defining how to display the applications: In the same window or in a new one.

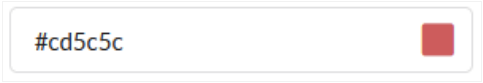
Icon Color

Defines the Font Awesome icon color. If no value is informed, the icon will receive the default color of the button theme.

The icon Color option is available whenever any Font Awesome option to set in the Display attribute.

You can set the icon color in the following ways.

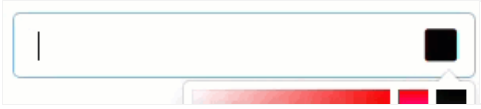
Informing a HEX code



Selecting a color

When selecting a color, the HEX value will be inserted into the field automatically.

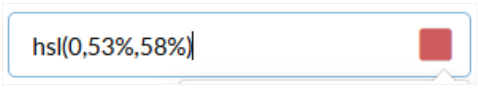
Informing the color name



When informing the color name, the HEX value will be inserted in the field automatically.

Informing a RGB code

Informing a HSL code



.png gif_informing_name: ../../../../assets/images/docs/app/menu/toolbar/informing_name.gif

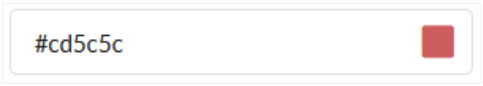
Hover Icon Color

Defines the Font Awesome icon color when hover in the button. If no value is informed, the icon will receive the default color of the button theme.

The Hover Icon Color option is available whenever any Font Awesome option to set in the Display attribute.

You can set the icon color in the following ways.

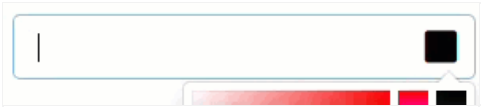
Informing a HEX code



Selecting a color

When selecting a color, the HEX value will be inserted into the field automatically.

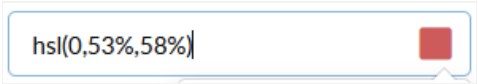
Informing the color name



When informing the color name, the HEX value will be inserted in the field automatically.

Informing a RGB code

Informing a HSL code



.png gif_informing_name: ../../../../assets/images/docs/app/menu/toolbar/informing_name.gif

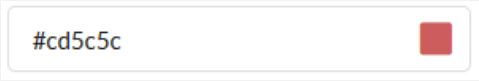
Disabled Icon Color

Defines the Font Awesome icon color when the button is disabled. If no value is informed, the icon will receive the default color of the button theme.

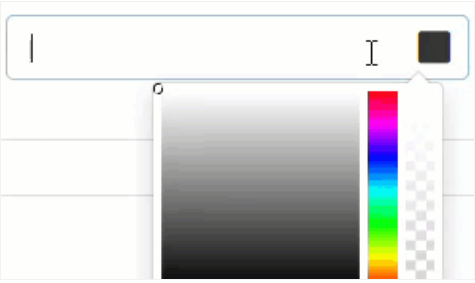
The Disabled Icon Color option is available whenever any Font Awesome option to set in the Display attribute.

You can set the icon color in the following ways.

Informing a HEX code

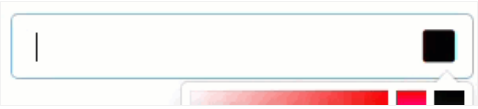


Selecting a color



When selecting a color, the HEX value will be inserted into the field automatically.

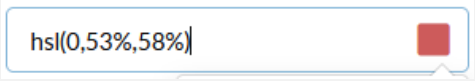
Informing the color name



When informing the color name, the HEX value will be inserted in the field automatically.

Informing a RGB code

Informing a HSL code



CSS Style

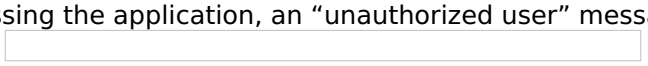
Define the style of the buttons

Related Links

Related Videos

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application. 

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Headers

Disable XSS Auditor

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

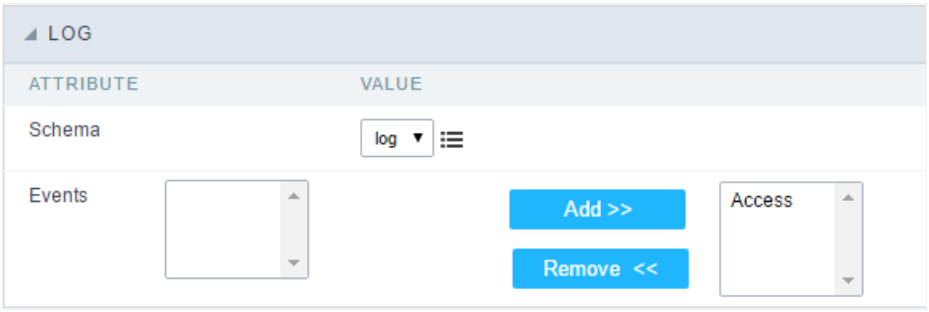
Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Menu Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Menu Events Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application's CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event's PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Menu

This event occurred before the application execute the SQL, and execute only once. Is used to do verification of variables, and security verification.

onExecute - Menu

This event works only in the "menu" applications and are executed when an application is called through a link of the menu. Normally are used to the decision taking before the application execution named sc_script_name.

onLoad - Menu

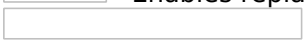
This event occurs when loading a horizontal / vertical menu. In this event we can run some security policy, for example if there is an attempt to access the menu directly, without first going through a login form, we can test a session variable defined in the login form, redirect the flow to the initial application. if(!isset([var_id_user]) or empty([var_id_user])) { sc_redirect("login.php"); }

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

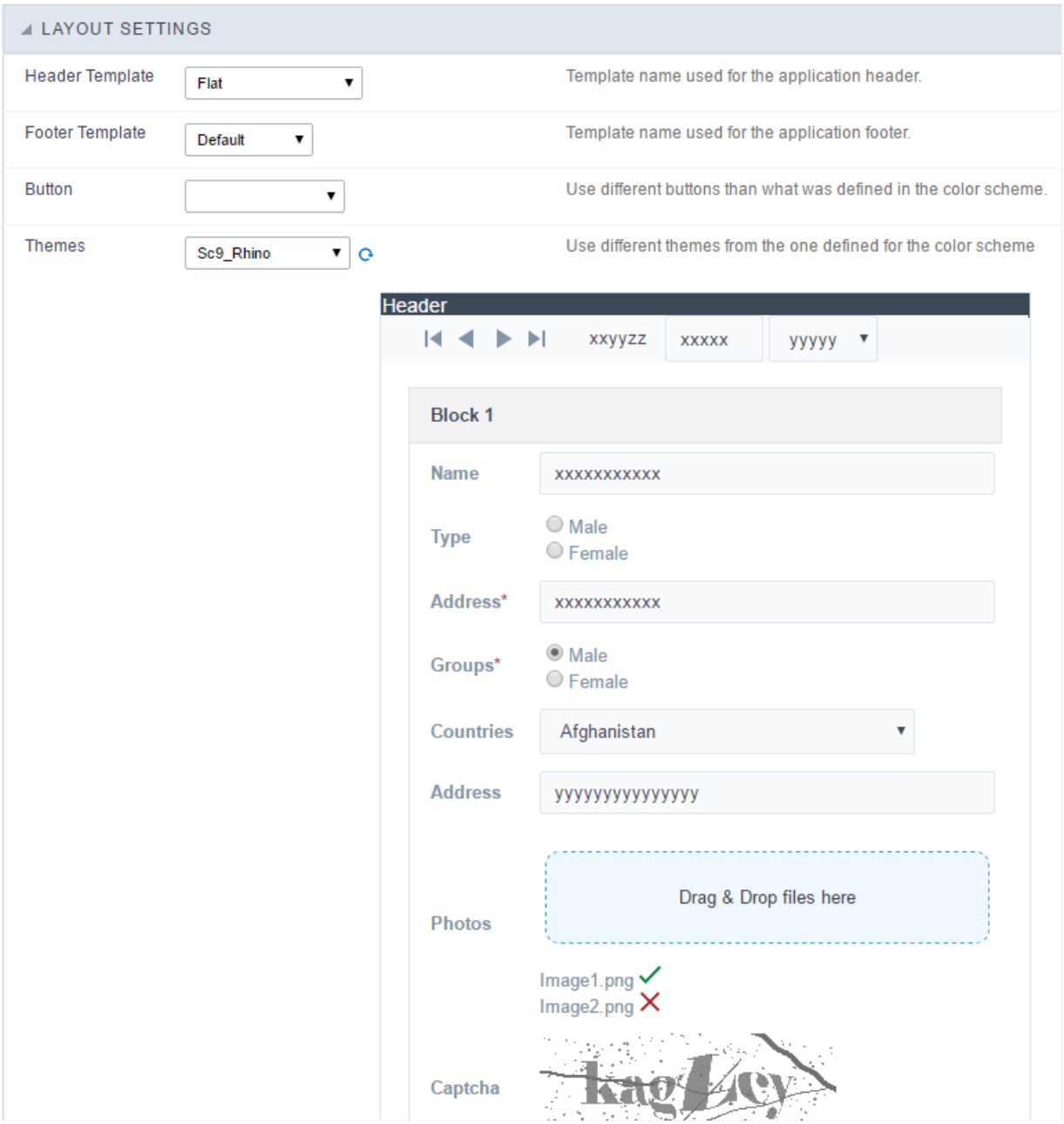
> Click Here to view the Scriptcase hotkeys documentation.

Menu Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout Settings

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

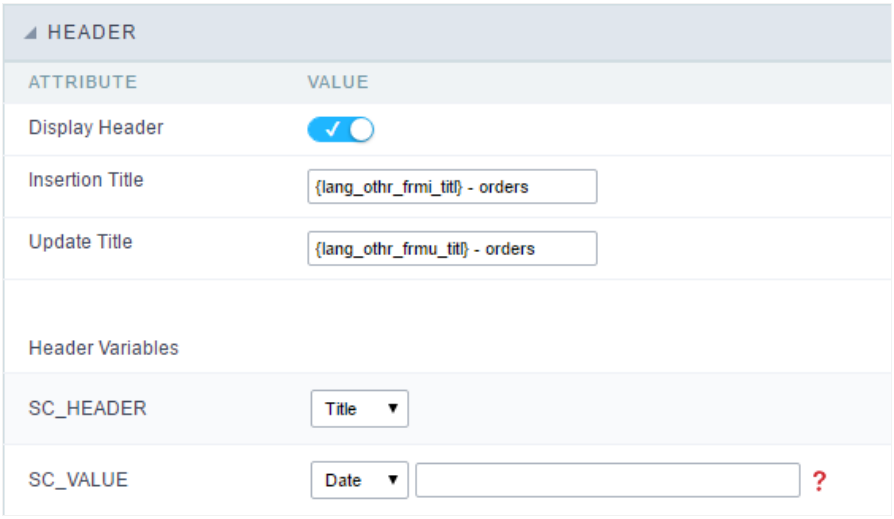
Allows choosing the button theme for the application.

Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

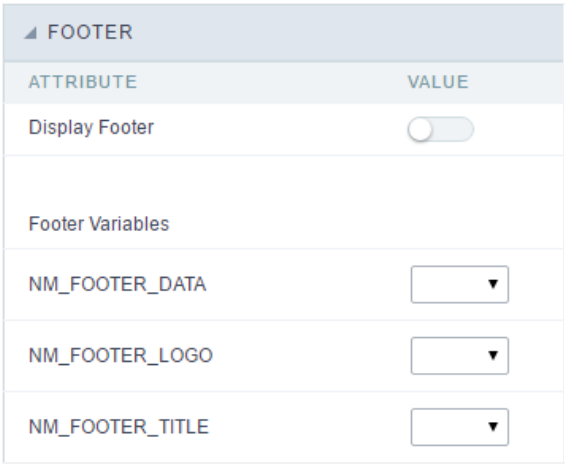
Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

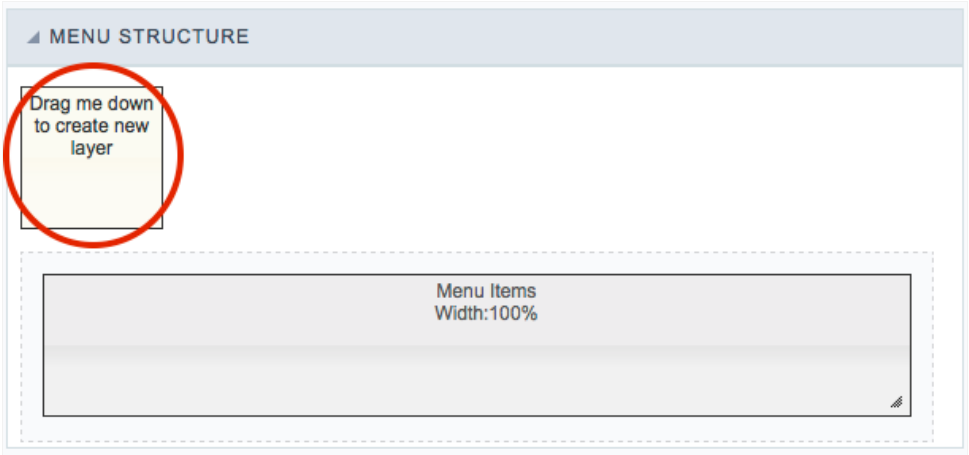
Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Menu Structure

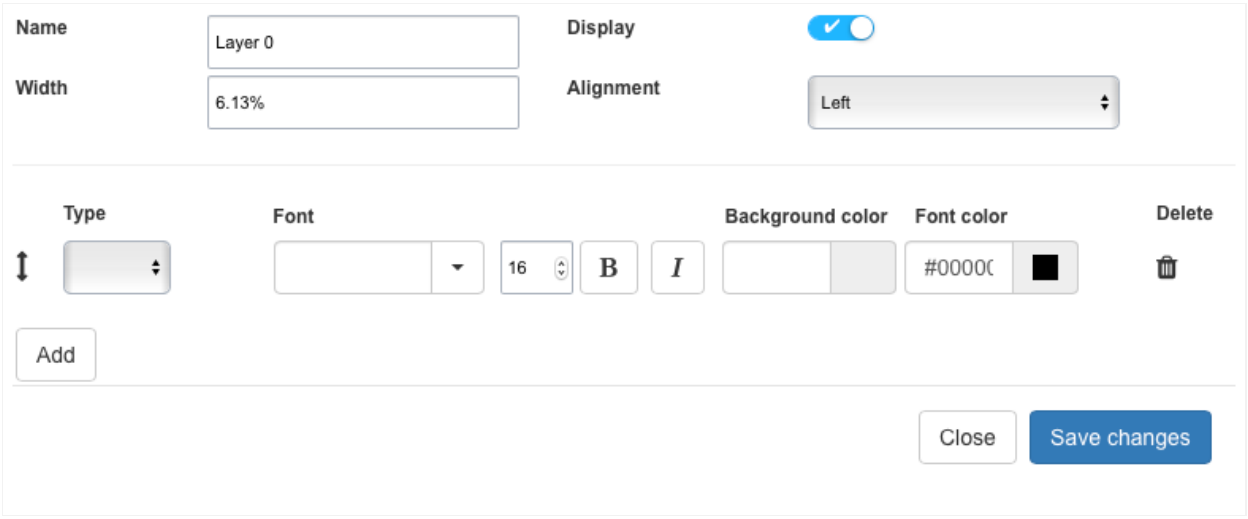
Using this option you can change the entire menu structure.



Using this option, you can change the entire menu structure. Drag the selected option to the "Menu Items" area to create a new "Layer" where you can add the contents. You can increase or decrease the layer width and also change its positioning within the "Menu items" area.

After adding the layer and customize its size, you can click on the “pencil” icon to edit the layer options or click on the “trash” icon to delete it.

Layer editing



Name

Use this option to set a name for the layer.

Width

It changes the layer size by using percentages or pixels. You need to change the “Menu Width” option within the Menu “Settings”. This value is in percentage as default.

Display

This option sets whether to display the layer or not.

Alignment

Using this option, you can change the alignment of all layer contents in the Left, Right, or Center.

Type

Using this option, you can select the data type of each row of the layer.

*__Title__: It displays the application title, according to the value set on “Application Title” within the menu “Header & Footer”.

*__Date__: It displays a date using your preferred date format. Example: d-m-Y

*__Image__: Add an image to the menu. Use the link to access the Image Manager from Scriptcase to choose an image or upload a new one.

*__Value__: It displays the content of the text input. You can inform static texts and “__Global Variables__” such as the logged user, for example: [usr_login].

*__Library__: Select an External Library from Scriptcase. You can import or create new libraries using the option “Tools » External Libraries” from the Scriptcase main menu.

*__Method__: Select an application method to apply to the layer. Create new methods by using the option “PHP Programming” within the menu “Programming.”

Font

Select one of the available font families from the list. Change the font size and set the text as bold and/or italic.

Background color

Change the background color of each row of the layer content.

Font color

Change the color for the text in the layer.

Delete

It allows deleting the layer line.

Add

It allows adding another line to the layer.

Close

It closes the edit screen of the layer.

Save

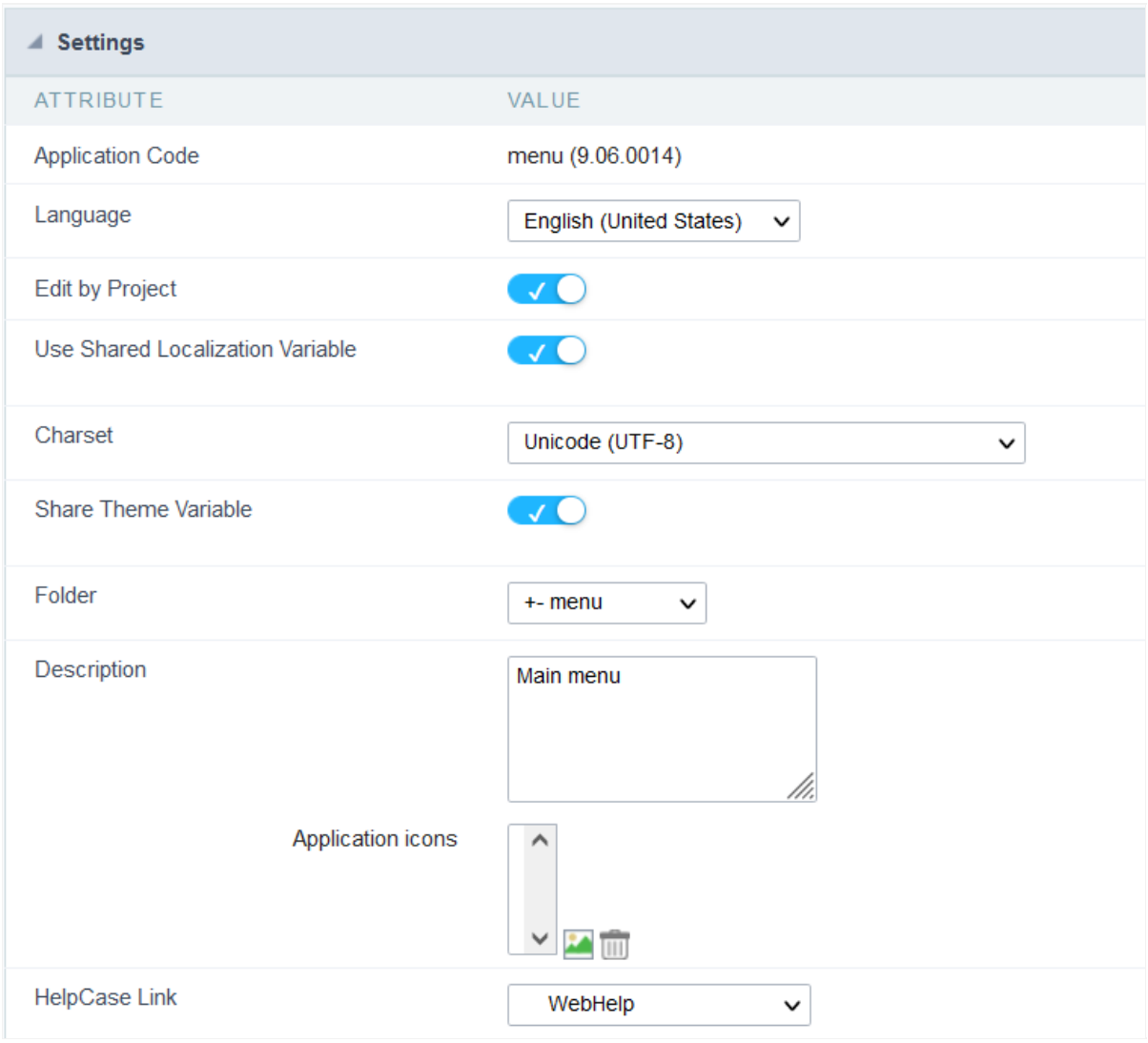
Save all changes for the layer.

Related Links 🔗

Related Videos ▷

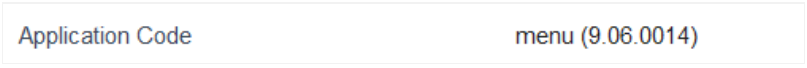
Application Settings (Menu)

Settings



Application Code

Informs the current name of the application and the version in which it was created



Language

Defines a specific language for the application, overriding the default language defined in the project. If no language is selected, the application will inherit the project's default language.

The languages listed in the option are defined in the project properties

Editing by Project

By disabling this option, only the user who created the application will have access to its settings.

No other project users will be able to access the application.

In the image below, the menu application has the edit by project option disabled.

Session Location

Defines if the language of the applications will be stored

Charset

Defines an application-specific charset.

If no value is selected, the application inherits the project's default charset.

Ideally, this charset option, within the application's settings, does not have a defined value. Different usage of your base charset can cause problems saving information in your tables.

Schemes in Session

When active, this option allows the theme stored in the session to be used by the application.

For example, by selecting theme x at login, all applications in the project will use the selected theme.

If the option is unchecked, the application remains with its original theme.

Folder

Folder where the application is stored in Project Explorer.

Description

Displays the description of the application.

You can edit or add a description also in the Project Explorer interface.

Extra Application Icons

It stores the images that will be used in the application through codes in the events, causing these images to be sent along with the application at the time of publication.

Only images used in codes, such as creating a dynamic menu with icons or creating a bill of exchange using a blank, need to be added.

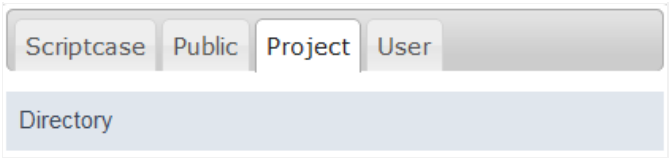
Images added in HTML Image fields or in the application header, for example, are already sent with the applications.

The added images will be stored in this directory: **../_lib/img/**

How to use the images

When inserting the image in the image manager, its name is changed according to the **scope** (public, project or user) and the **storage directory** (background, button, icon, menu or general) following the following pattern.

Scope: It refers to the level of access to the file within Scriptcase, in the development environment.

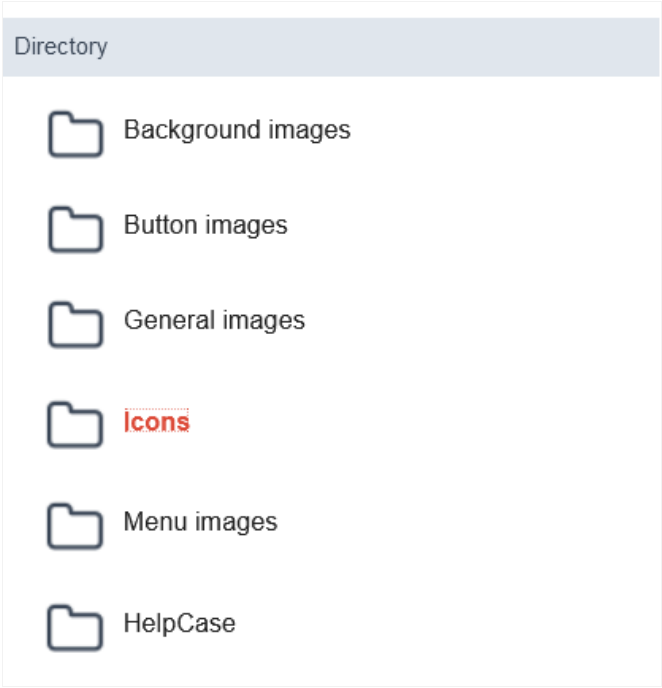


- Scriptcase__ - Contains preexisting images in the tool, where it is not possible to upload new images.
- Public__ - Images accessible for all projects.
- Project__ - Images accessible only in the project in which it was inserted.
- User__ - Images accessible only to the user who uploaded the image.

Each scope adds a prefix to the image name.

- Public - **sys__NM__**
- Project - **grp__NM__**
- User - **usr__NM__**

Storage Directory: Directory where the image will be added. Each directory adds one more prefix to the image name.



- background images - **bg__NM__**
- button images - **btn__NM__**
- general images - **img__NM__**
- icons - **ico__NM__**
- menu images - **menu_img__NM__**

This way, an image called _001.jpg, added in the public scope and in the icon directory will be named **sys__NM__ico__NM__001.jpg**

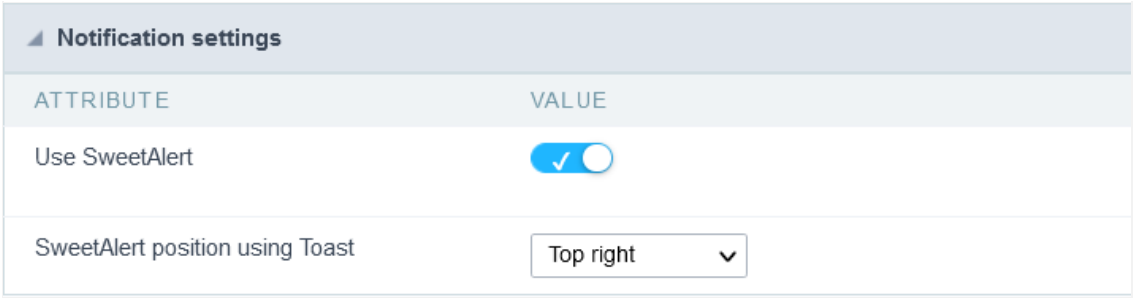
HelpCase Link

Associates HelpCase files to your application.

This configuration can be done expressly in the HelpCase settings, where it is possible to change this option for all project applications. See how by clicking here.

The HelpCase button must be selected in the toolbar to be able to access the selected page.

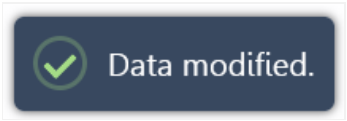
Notifications Settings



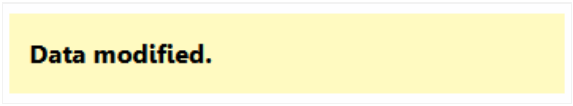
Use SweetAlert

Activate SweetAlert in the application, replacing the “confirm” and “alert” of the browser.

Confirm Example with SweetAlert Enabled

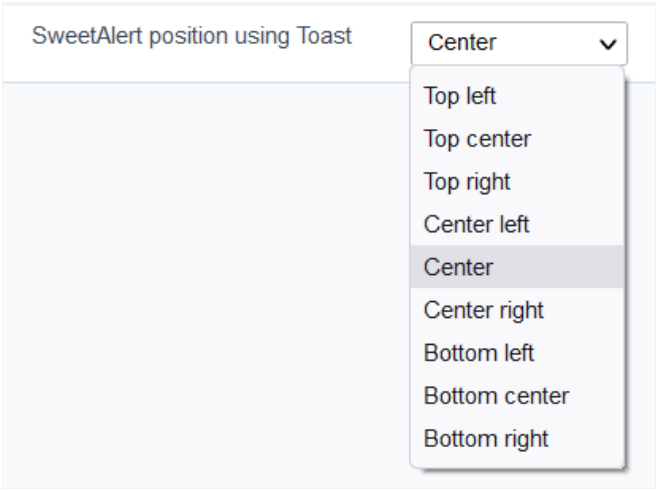


Confirm Example with SweetAlert Disabled

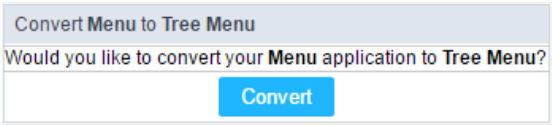


SweetAlert position using Toast

Sets the display position of the message window using toast.



Convert menu type



Scriptcase makes available two types of menu application, they are: Tree Menu and Menu(Horizontal).

This tool allows the conversion of an application from **Menu** to **Menu Tree** or from **Menu Tree** to **Menu**, keeping header and item settings as well as event codes.

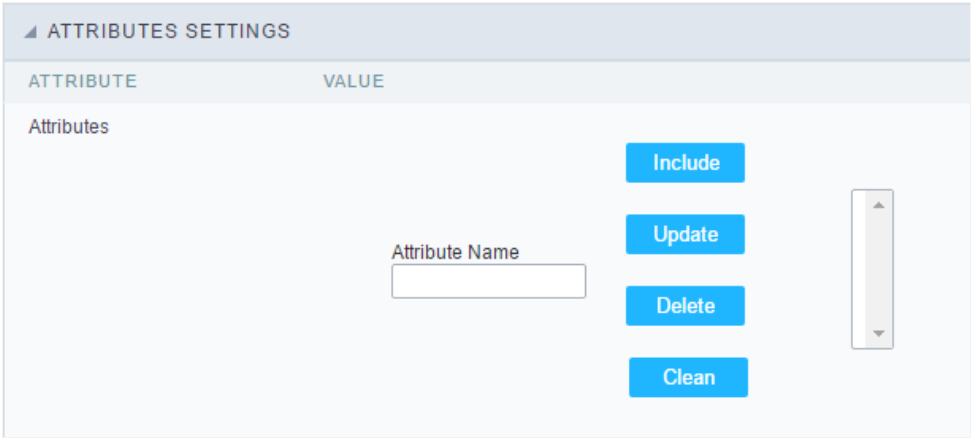
The menu conversion only changes its format, keeping all the settings of the original application.

Menu Programming

ScriptCase has incorporated the concept of Object Oriented programming, using attributes, resources, methods and libraries. It is possible to create your own business rules in applications, and by using these concepts you can reap huge rewards in terms of better organization and improved development.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

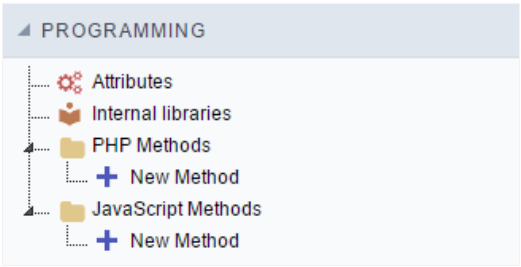
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

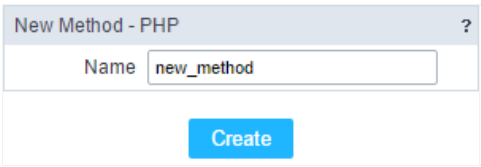
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

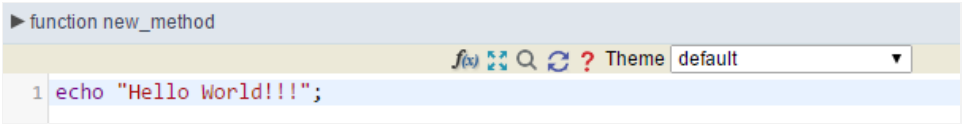


- Give the method a name and click Create. As in the image below.

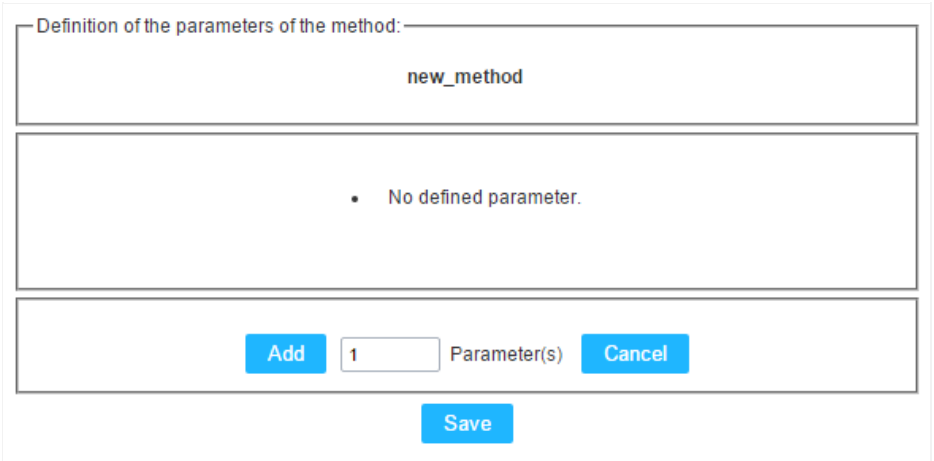
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



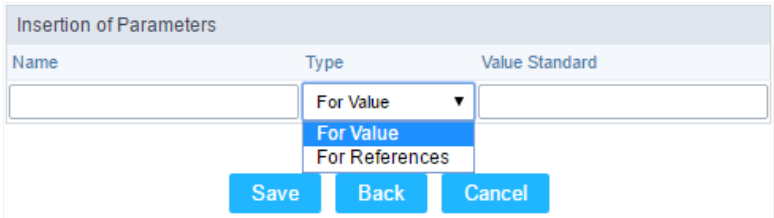
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

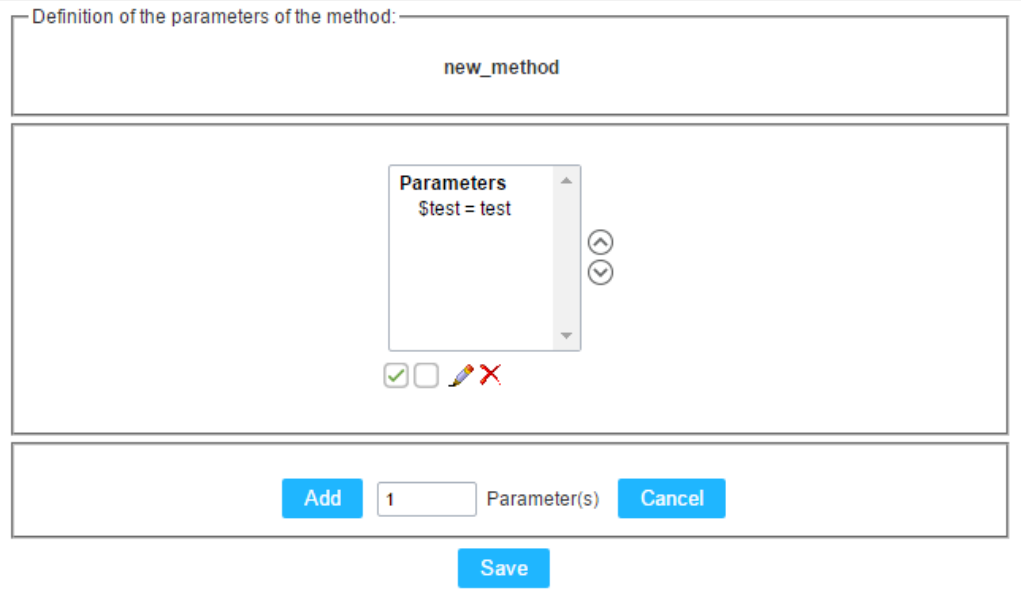


- **Name** : Type in the variable's name.
- **Type** : Selecting the type of variables: For Value or For Reference.

- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:



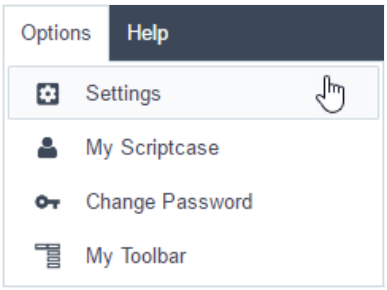
- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

Creating a Menu Application

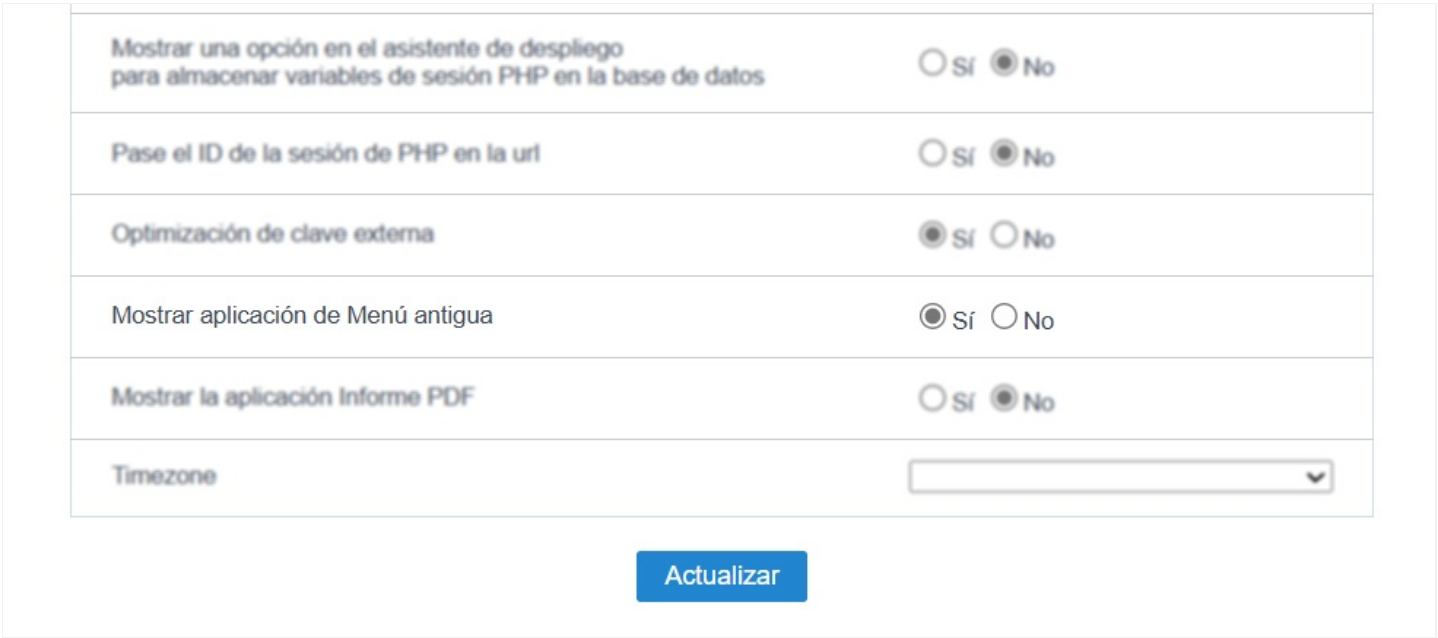
New Application

With the implementation of the responsive menu, the old menu application was removed from the interface. However, it is still possible to use it, keeping in mind that it does not include some features developed exclusively for the responsive menu.

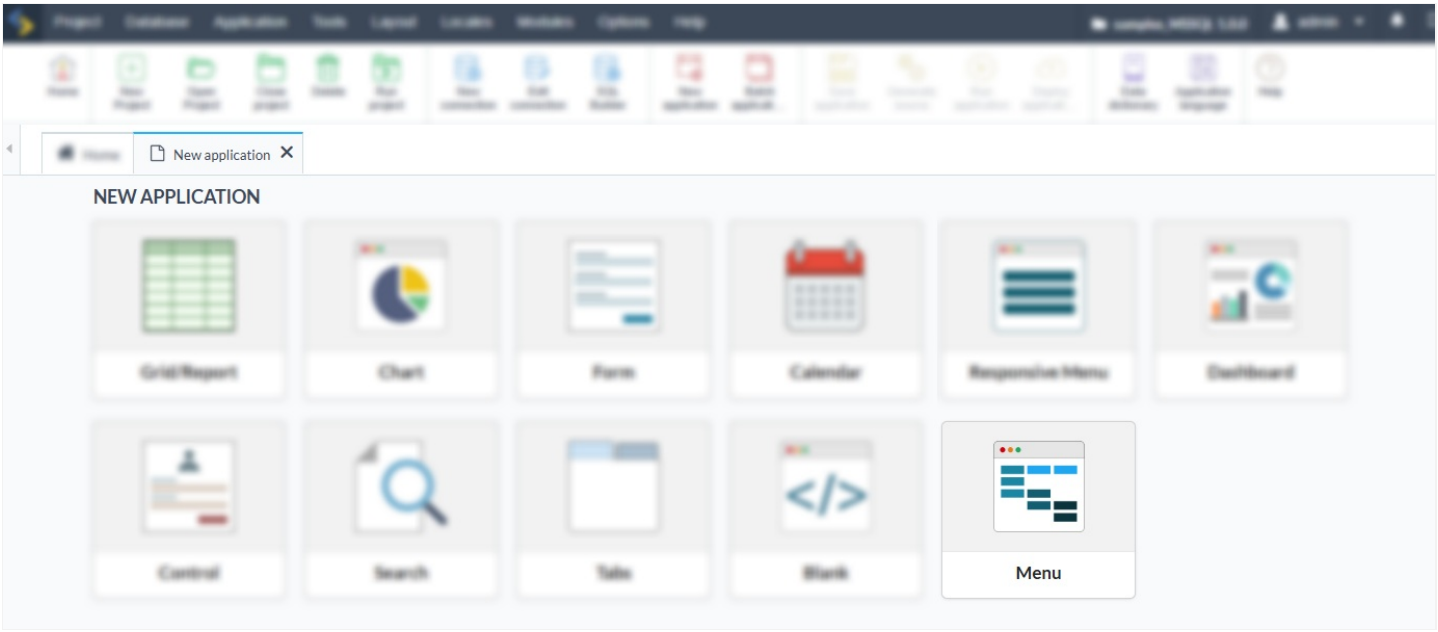
To enable this functionality, the Scriptcase administrator must access the menu Options > Settings and open the System Settings page.



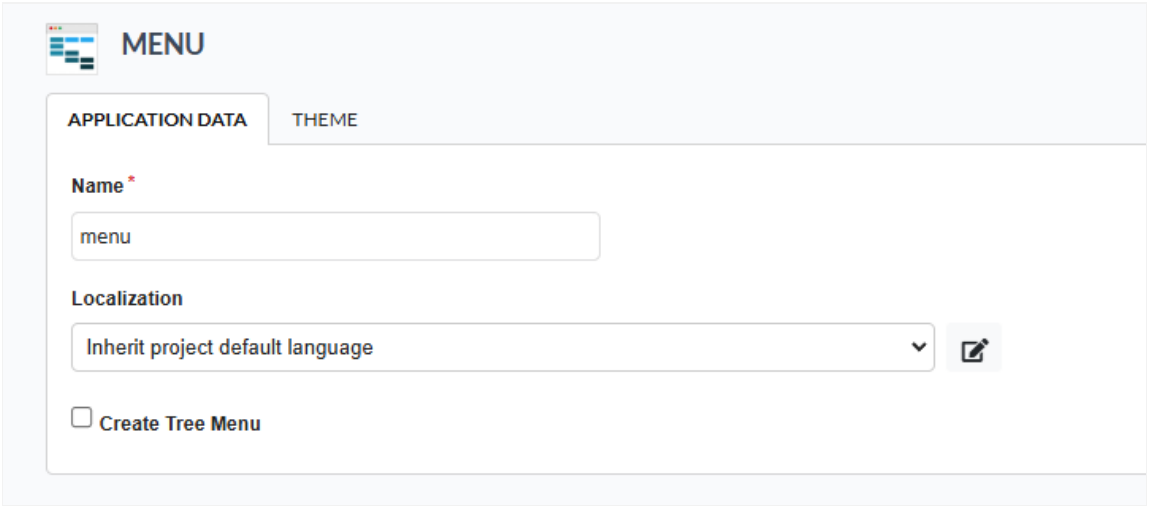
At the bottom of the page, after the foreign key configuration section, you can enable the old menu application to display it as an option when creating new applications.



Once enabled, the application will appear on the new application creation screen, as shown in the image below.



Application Data



Name

The name of the application being created. It cannot contain special characters.

Localization

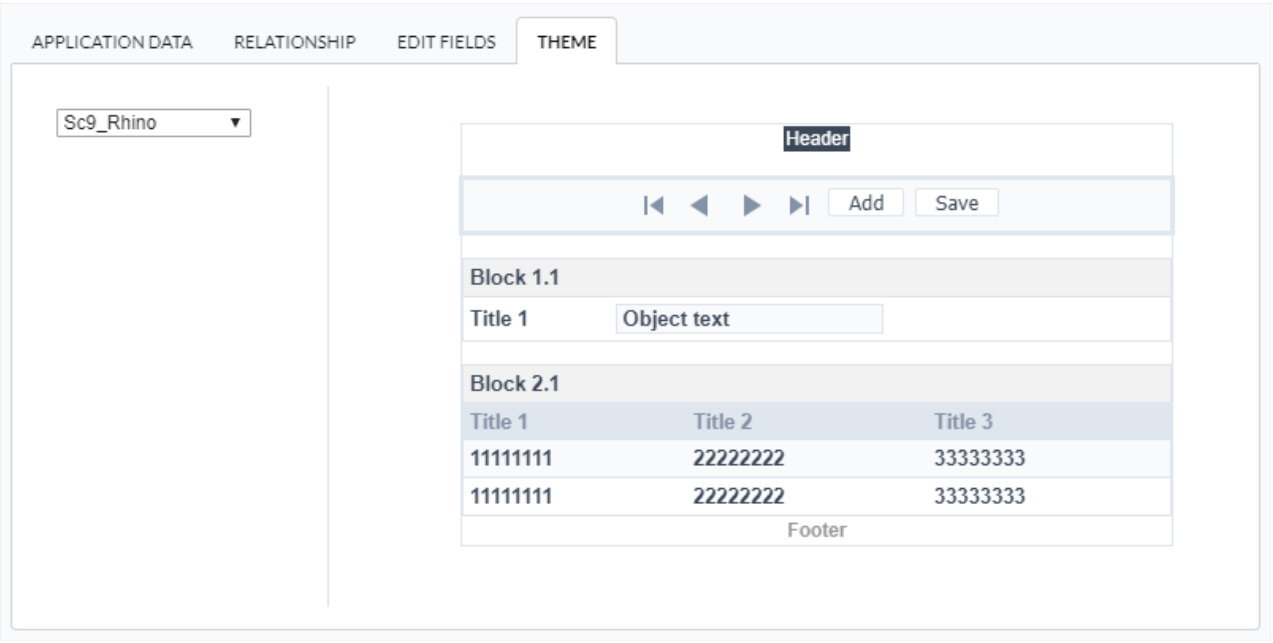
The language of the application to be created. The project's default language is automatically selected.

Create Tree Menu

Allows you to create a menu application in a tree format, where the elements are organized hierarchically with expandable options.

Theme

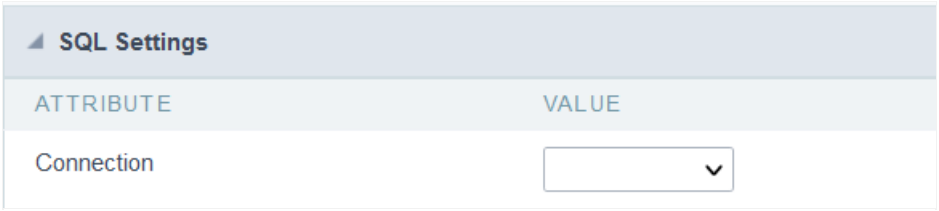
On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.



Related Links 🔗

Related Videos ▷

Menu SQL Settings



The Connection field defines the connection that will be used by the application. In Menu type application, it is not mandatory to select a connection.

By default, the connection field has no value after creating a menu type application.

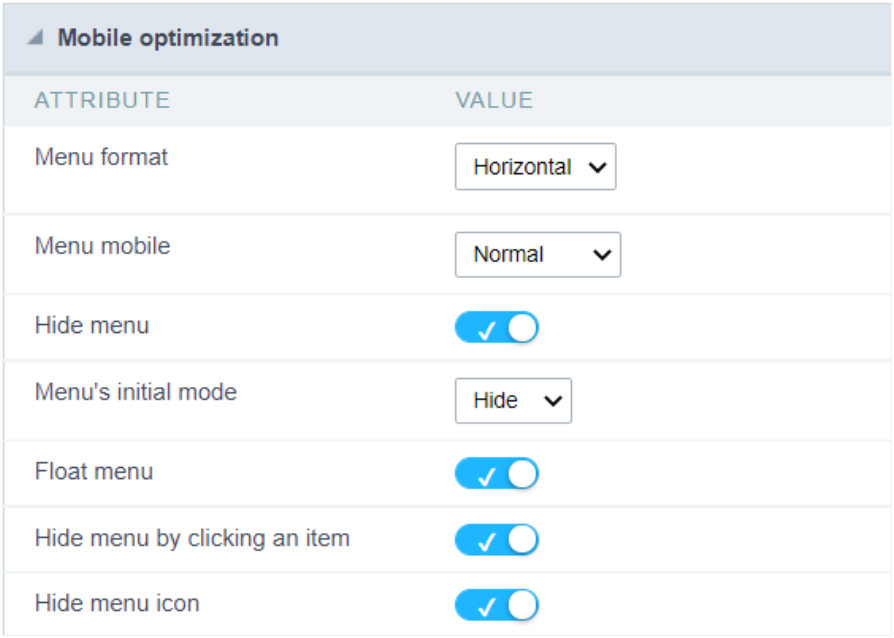
In menu applications, it is only necessary to select a connection to use some code in the events, for example, to create a dynamic menu.

Menu Mobile Optimization

The Mobile settings allows to define the automatic optimization of generated applications to run on mobile devices.

See below the available settings.

Mobile optimization

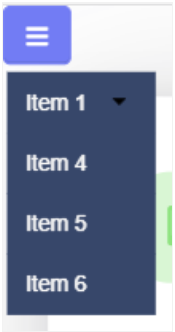


Menu Format

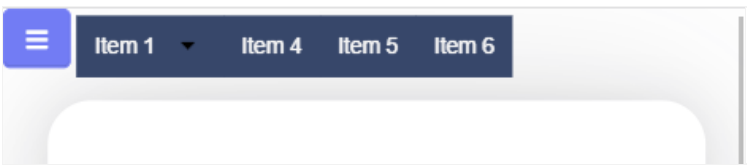
Sets the menu items orientation when the **Menu mobile** option is set to **normal** value

The menu can be displayed vertically or horizontally. See the examples below.

Vertical menu example



Horizontal menu example



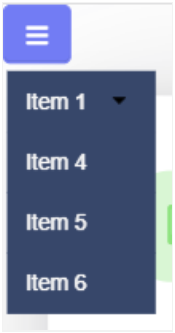
Mobile Menu

Sets the menu display format in two ways: **normal** and **navigation**

Normal

Sets the menu display format in two ways: **normal** and **navigation**

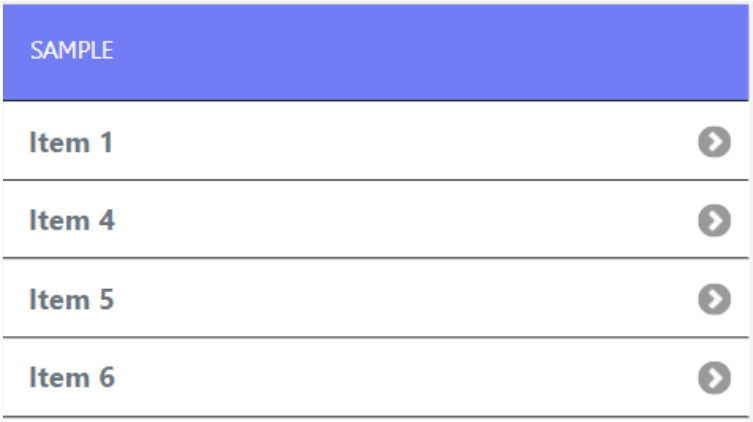
Sample normal menu



Navigation

This type of menu expands and takes up the entire screen, so that navigation is limited to only the options of the selected item.

Sample Navigation Menu



hide the menu

Defines whether the menu can be collapsed or will remain fixed on the screen.

When **enabled**, this feature allows the end user to hide the menu. If the developer chooses to **disable this option**, the menu items will continue to be displayed on the screen.

The behavior of the hide menu option is defined in the detailed options below and are enabled only when activating the hide menu feature.

Initial state of the menu

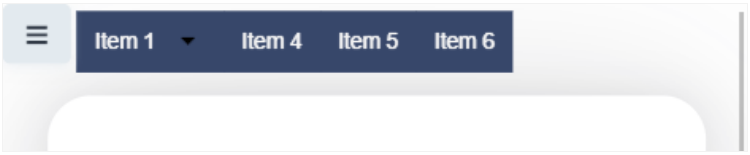
Only available when the **Hide menu** option is enabled. This setting defines the initial state of the menu.

The options are **Open** and **Hide**.

Open

This value displays the menu items on the initial access to the application, without the need to click on the button to expand the menu.

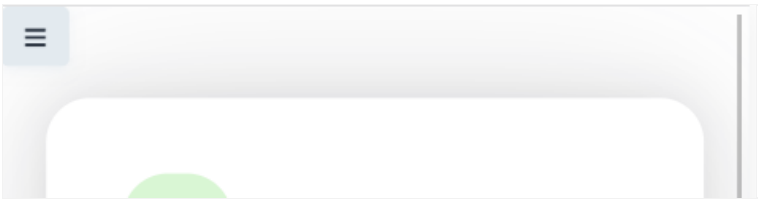
Example of open starting menu



Hide

In this option the menu items will not be displayed in the initial load of the application, it is necessary to create the button to expand the menu to perform the navigation.

Example of menu starting closed



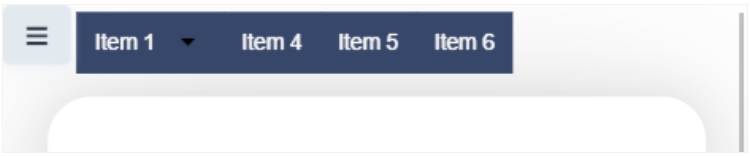
float menu

Defines how the items are opened in relation to the application.

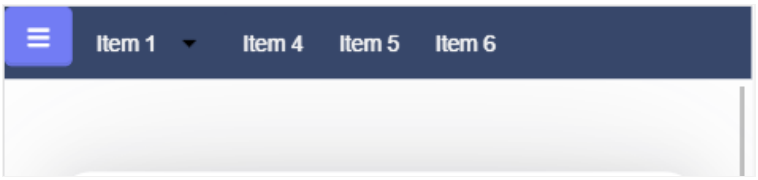
When enabled, items will be displayed floating, overlaying the application without changing its display.

When disabled, items will be opened together with the application, occupying space and moving it away during the period it is open.

Example with option enabled



Example with the option disabled



Hide menu when clicking an item

This feature allows menu items to be collapsed when clicking on one of the items to open an application.

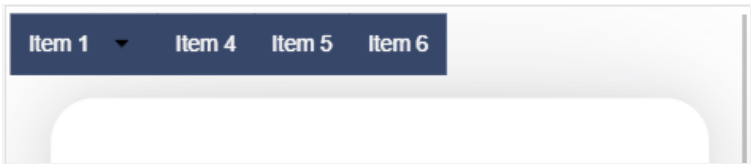
Hide menu icon

Defines whether the Hide Menu button will always be visible.

This option is only available if the **Hide menu when clicking an item** is enabled.

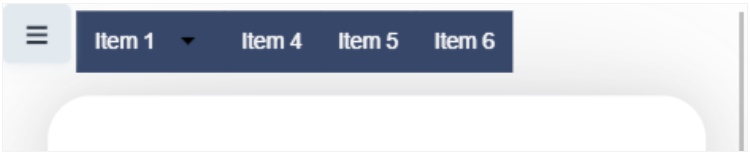
Enabled

In this case, the hide menu button will also be hidden when clicked, returned to display as soon as an item is clicked to open an application.



Disabled

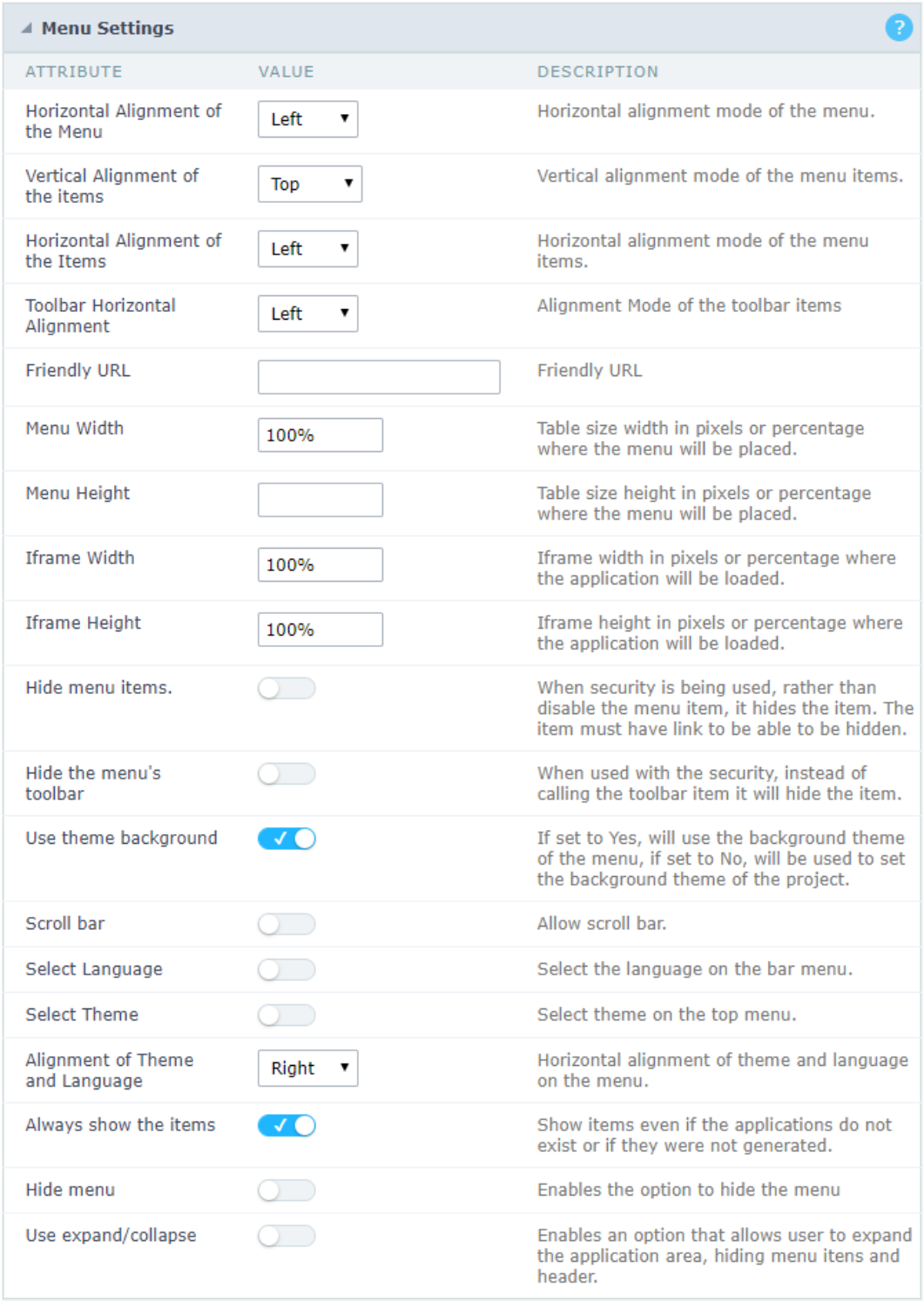
In this case, the hide menu button will always be visible, even when items are being displayed.



Tree Menu Settings

Desktop

The menu settings allow to define the attributes of the application display, according to the following:



Menu Settings

ATTRIBUTE	VALUE	DESCRIPTION
Horizontal Alignment of the Menu	Left	Horizontal alignment mode of the menu.
Vertical Alignment of the items	Top	Vertical alignment mode of the menu items.
Horizontal Alignment of the Items	Left	Horizontal alignment mode of the menu items.
Toolbar Horizontal Alignment	Left	Alignment Mode of the toolbar items
Friendly URL		Friendly URL
Menu Width	100%	Table size width in pixels or percentage where the menu will be placed.
Menu Height		Table size height in pixels or percentage where the menu will be placed.
Iframe Width	100%	Iframe width in pixels or percentage where the application will be loaded.
Iframe Height	100%	Iframe height in pixels or percentage where the application will be loaded.
Hide menu items.		When security is being used, rather than disable the menu item, it hides the item. The item must have link to be able to be hidden.
Hide the menu's toolbar		When used with the security, instead of calling the toolbar item it will hide the item.
Use theme background	✓	If set to Yes, will use the background theme of the menu, if set to No, will be used to set the background theme of the project.
Scroll bar		Allow scroll bar.
Select Language		Select the language on the bar menu.
Select Theme		Select theme on the top menu.
Alignment of Theme and Language	Right	Horizontal alignment of theme and language on the menu.
Always show the items	✓	Show items even if the applications do not exist or if they were not generated.
Hide menu		Enables the option to hide the menu
Use expand/collapse		Enables an option that allows user to expand the application area, hiding menu itens and header.

Menu configuration Interface.

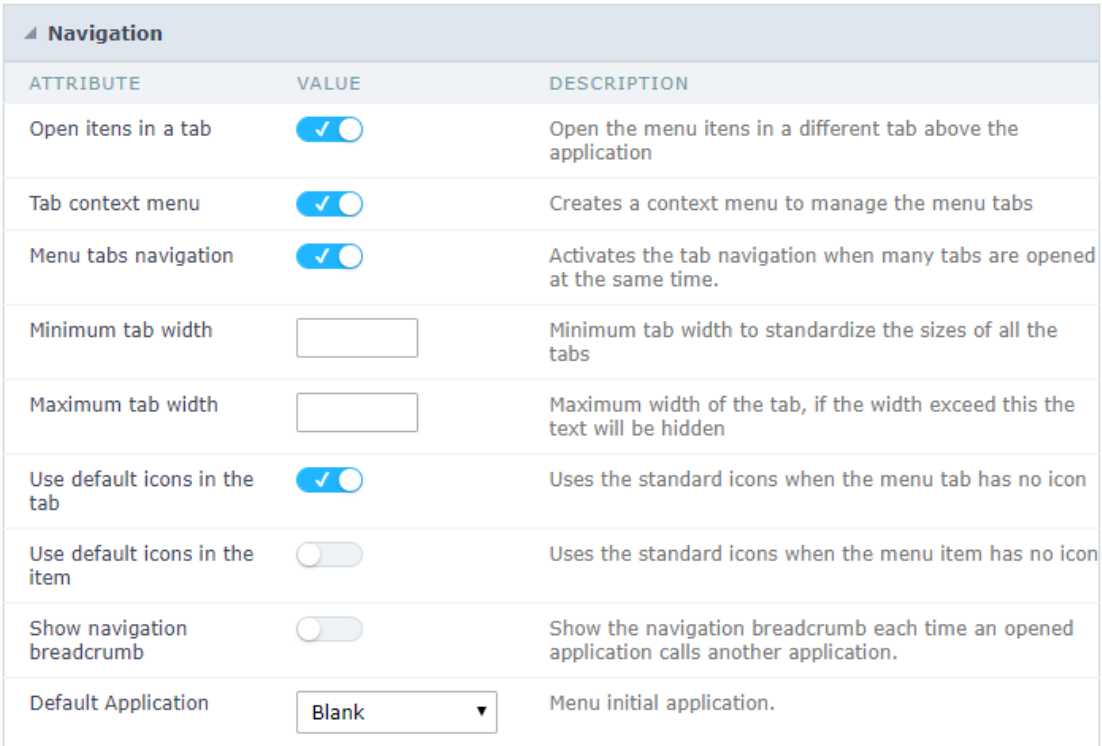
- **Horizontal Alignment of the Menu**: Defines the horizontal alignment of the menu (Left, Center, Right).
- **Vertical Alignment of the items**: Defines the vertical alignment of the menu items (Top, Center, Bottom).
- **Horizontal Alignment of the Items**: Defines the horizontal alignment of the menu items (Left, Center, Right).
- **Toolbar Horizontal Alignment**: Defines the horizontal alignment of the toolbar items (Left, Center, Right).
- **Friendly URL**: Allows defining a name for the URL that's going to be called by the application. The characters allowed in the URL are(a-z, A-Z, 0-9). You still can inform the Friendly URL on the project home, where there is a list of applications with

the “Friendly URL” column.

- **Menu Width**: Define the width of the menu in pixels or percentages.
- **Menu Height**: Define the height of the menu in pixels or percentages.
- **Iframe Width**: Define the width of the Iframe (Where the applications will open in) in pixels or percentages.
- **Iframe Height**: Define the height of the Iframe (Where the applications will open in) in pixels or percentages.
- **Hide Menu Items**: This attribute defines whether the menu item will be displayed or not. When enabled, items are hidden if they do not have a link, the application has active security (without user access), or the item is disabled using the `sc_menu_disable` macro. When disabled, the items will always be displayed.
- **Hide the menu’s toolbar**: When enabled and used with the security module, it’ll hide the toolbar that the user doesn’t have permission to access, instead of only disabling them.
- **Use theme background**: If enabled, it’ll use the background theme of the menu. Else it’ll use the background defined in the project.
- **Scroll bar**: Enables the scroll bar in the Iframe.
- **Select Language**: Allows to select the language in the menu bar.
- **Select Theme**: Allows to select the theme in the menu bar.
- **Alignment of Theme and Language**: Defines the horizontal alignment of the language and theme comboboxes.
- **Always show the items**: Displays the application items, even though they don’t exist, or haven’t been generated yet.
- **Hide menu**: Enables the option to hide the menu.

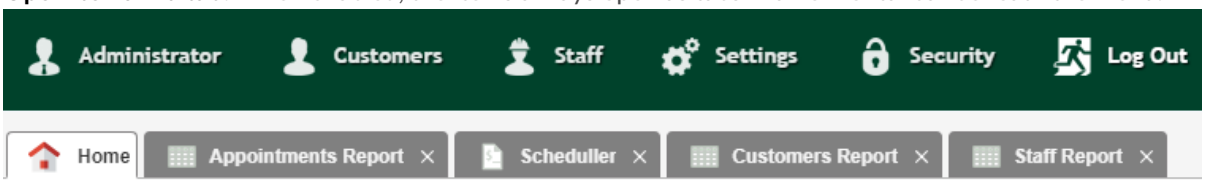
Navigation

The menu navigation settings allow to define the attributes of the tabs and the navigation path, according to the following:



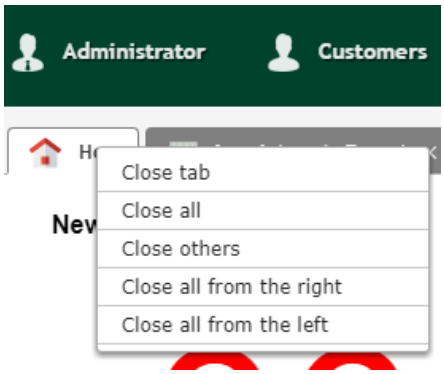
Menu Navigation configurations Interface.

- **Open items in a tab**: When enabled, the items always open as tabs in a horizontal bar beneath the menu.



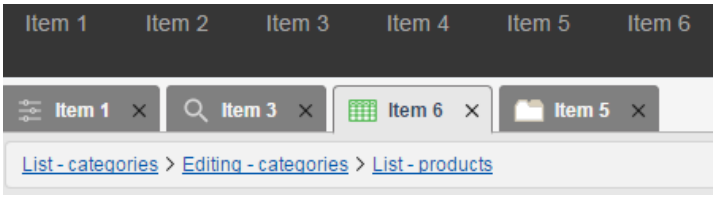
Navigating through Applications with tabs.

- **Context menu of the Tabs**: Allows you to manage the open tabs.



Application with a context menu.

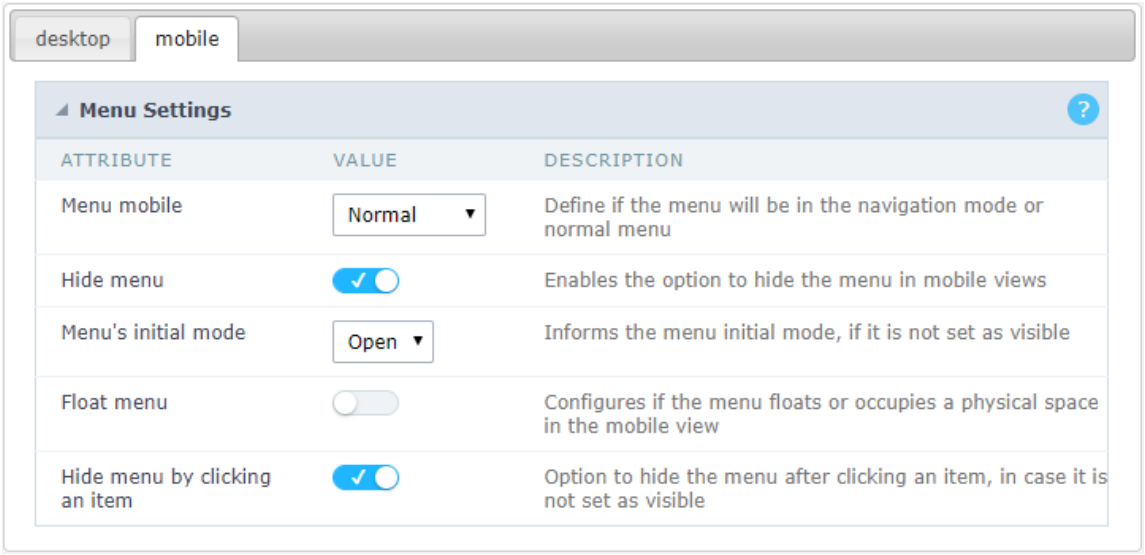
- **Minimum tab-width**: Allows to inform the min-width of all the tabs.
- **Maximum tab-width**: Allows to inform the max-width of all the tabs.
- **Show navigation breadcrumb**: Displays the path containing the application in the tab.



Navigation path Display in the Application.

- **Use default icons in the item**: When enabled, this option displays the standard icons for each item.
- **Use default icons in the tab**: When enabled, this option displays the standard icons for each tab.
- **Default Application**: Select a default application initiate when the menu opens.
- **Show the initial app as a tab**: Allows to open the default application in a tab.

Mobile



Mobile menu configuration Interface.

- **Hide menu**: Enables the option to hide the menu when viewing from a mobile device.
- **Menu's initial mode**: Inform the starter mode of the menu. Otherwise, it'll always be visible.
- **Hide menu by clicking on an item**: When enabled, it hides the menu when clicking on an item.
- **Float Menu**: Hides the icon to open the menu or always displays it.

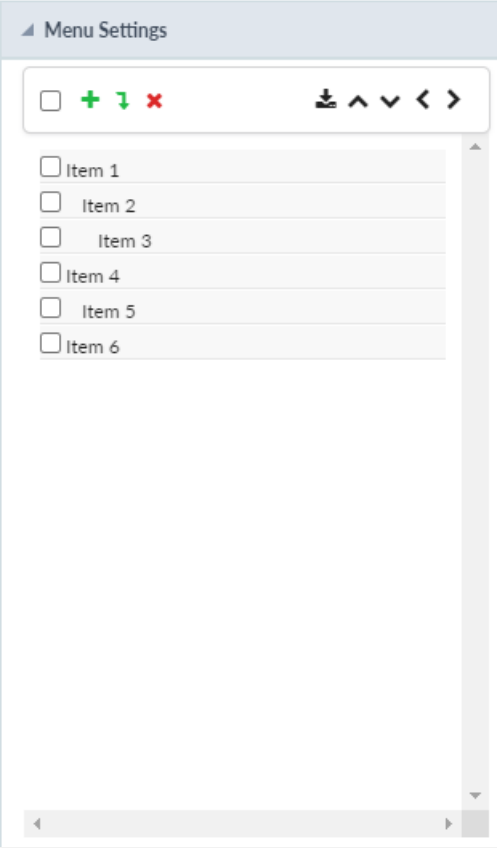
Related Samples 🔗

Related Videos ▷

Menu Items

Menu items settings

In this Screen the user have the settings to configure all the menu structure, below you will see how to use each setting.



Insert Item

With this button, the user can add a new item to the menu structure. Every time the user clicks in the button will generate a new item with the id.

Ex: Item 1

Insert Subitem

With this button, the user can add a new subitem in the menu structure. Every time the user clicks in the button will generate a new subitem but differently from the Insert item, when pressed will generate a subitem inside a subitem, so, to have all the items inside the same, the user must select the first item(The father item).

Remove Item

With this button, the user can delete an item or subitem by selecting it on the checkbox.

Import Applications

With this button, the user can select the applications want to use as menu items, this way the user doesn't need to create an item and then link the application to it.

Move Up
With this button, the user can move the position of the item up.

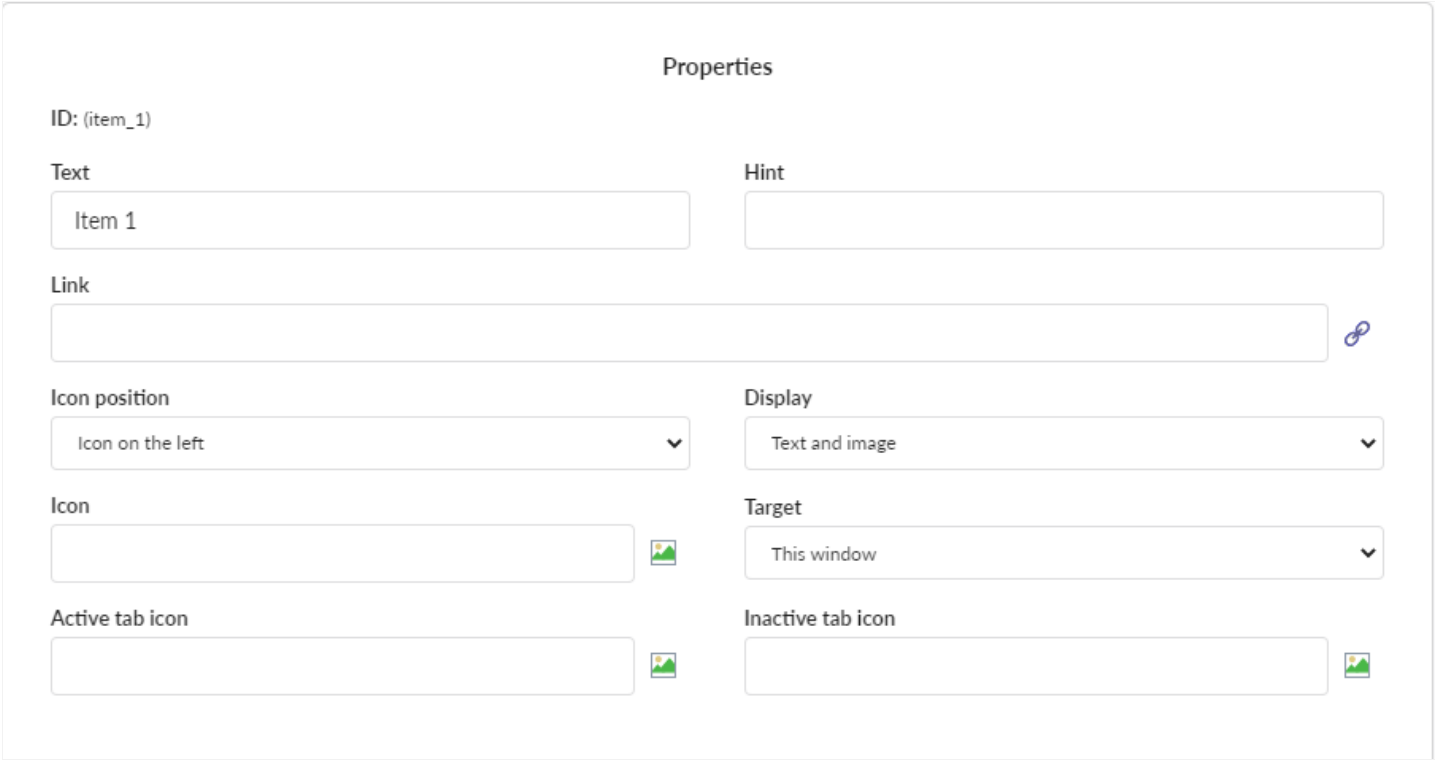
Move Down
With this button, the user can move the position of the item up.

Move Left
With this button, the user can move the position of the item left, this will make the item to be add as a subitem or not.

Move Right
With this button, the user can move the position of the item right, this will make the item to be add as a subitem or not.

Properties

In this settings, the user can configure how the item will look in the Menu.



ID
This option shows the actual item that is being edited.

Text
In this option the user configure the text that will be shown in the menu item.

Hint
In this option the user can configure a help text when the mouse is over the item.

Link
In this option the user configure the application that will be open when the link is clicked.

Display
In this option the user can configure how the item will be displayed. There are two options:

- Text and Image - Show the item with an imagem and a text side by side.

- Text and icon Font Awesome - Show the item with an icon from the Font Awesome library and a text side by side.

Icon
If in the display option is selected Text and Image here the user will select the image to be along with the text. If in the display option is selected Text and icon Font Awesome here the user will select the icon to be along with the text.

Active tab icon
This option allows the user to select the icon or image (same condition to the Icon settings from above) to be shown when the item is actived.

Inactive Tab icon
This option allows the user to select the icon or image (same condition to the Icon settings from above) to be shown when the item is deactivated.

Target
This option allows to select what will happens when the user clicks on the item. There are three options:

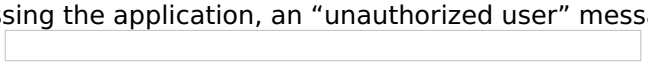
- This Window - Will open the application in the same window.
- New Window - Will open a new window for the application be displayed.
- Leave - Will exit the application, most used with the login application to return to the login.

Related Samples 🔗

Related Videos ▷

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application. 

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Headers

Disable XSS Auditor

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

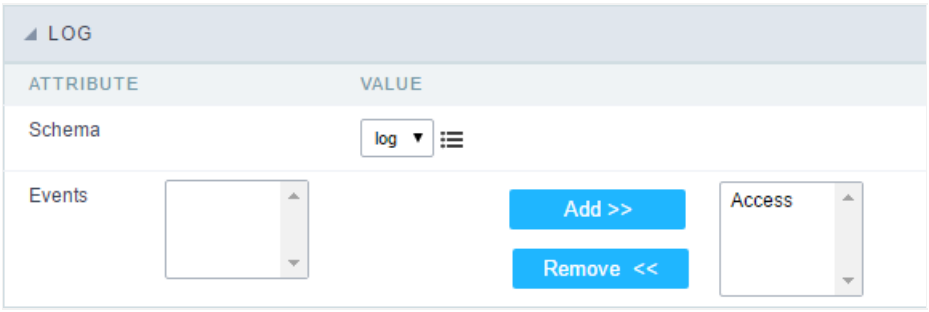
Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Tree Menu Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Tree Menu Evetns Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application’s CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event’s PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Tree Menu

This event occurred before the application execute the SQL, and execute only once. Is used to do verification of variables, and security verification.

onExecute - Tree Menu

This event works only in the "menu" applications and are executed when an application is called through a link of the menu. Normally are used to the decision taking before the application execution named sc_script_name.

onLoad - Tree Menu

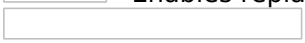
This event occurs when loading a horizontal / vertical menu. In this case, we can execute some security policy, for example, if you try to access the menu directly, without first going through a login form, we can test a session variable defined in the login form, redirect the flow to the initial application. if(!isset([var_id_user]) or empty([var_id_user])) { sc_redirect("login.php"); }

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

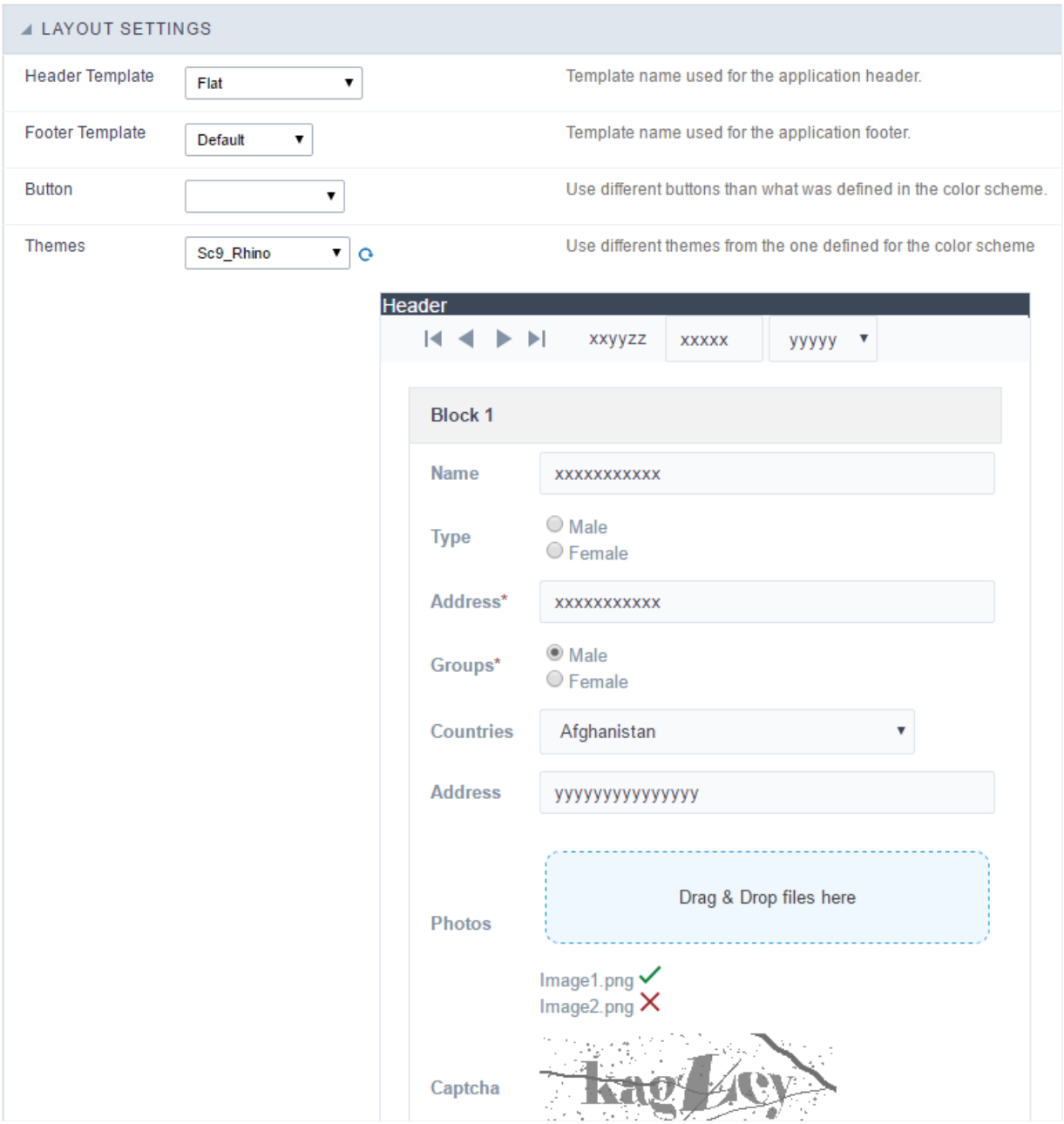
> Click Here to view the Scriptcase hotkeys documentation.

Tree Menu Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout Settings

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

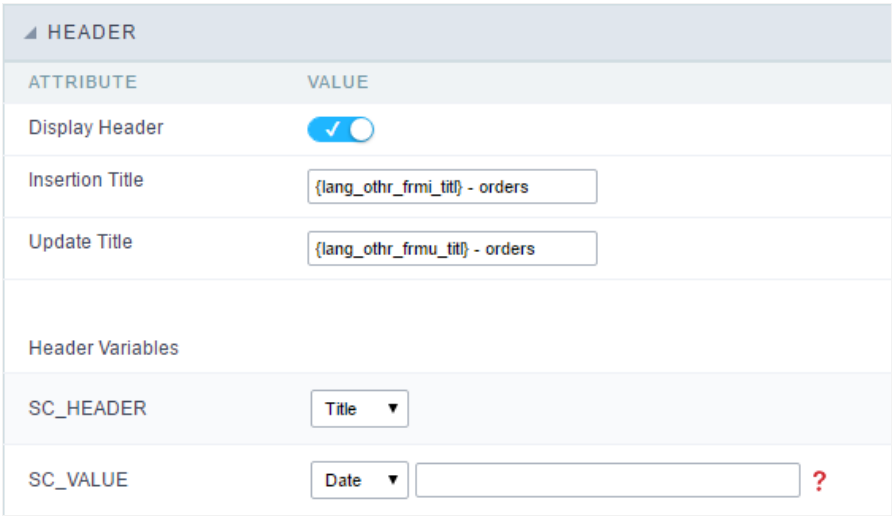
Allows choosing the button theme for the application.

Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

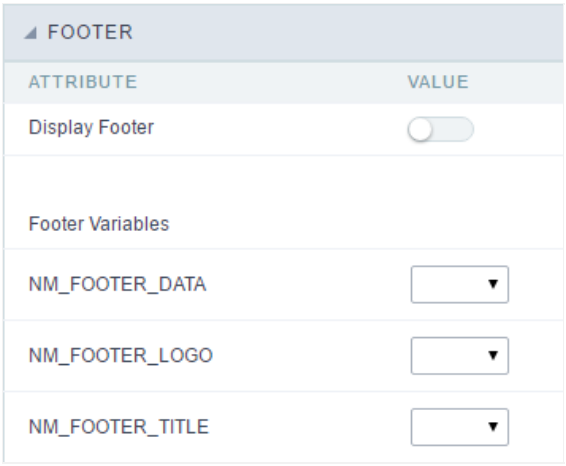
Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

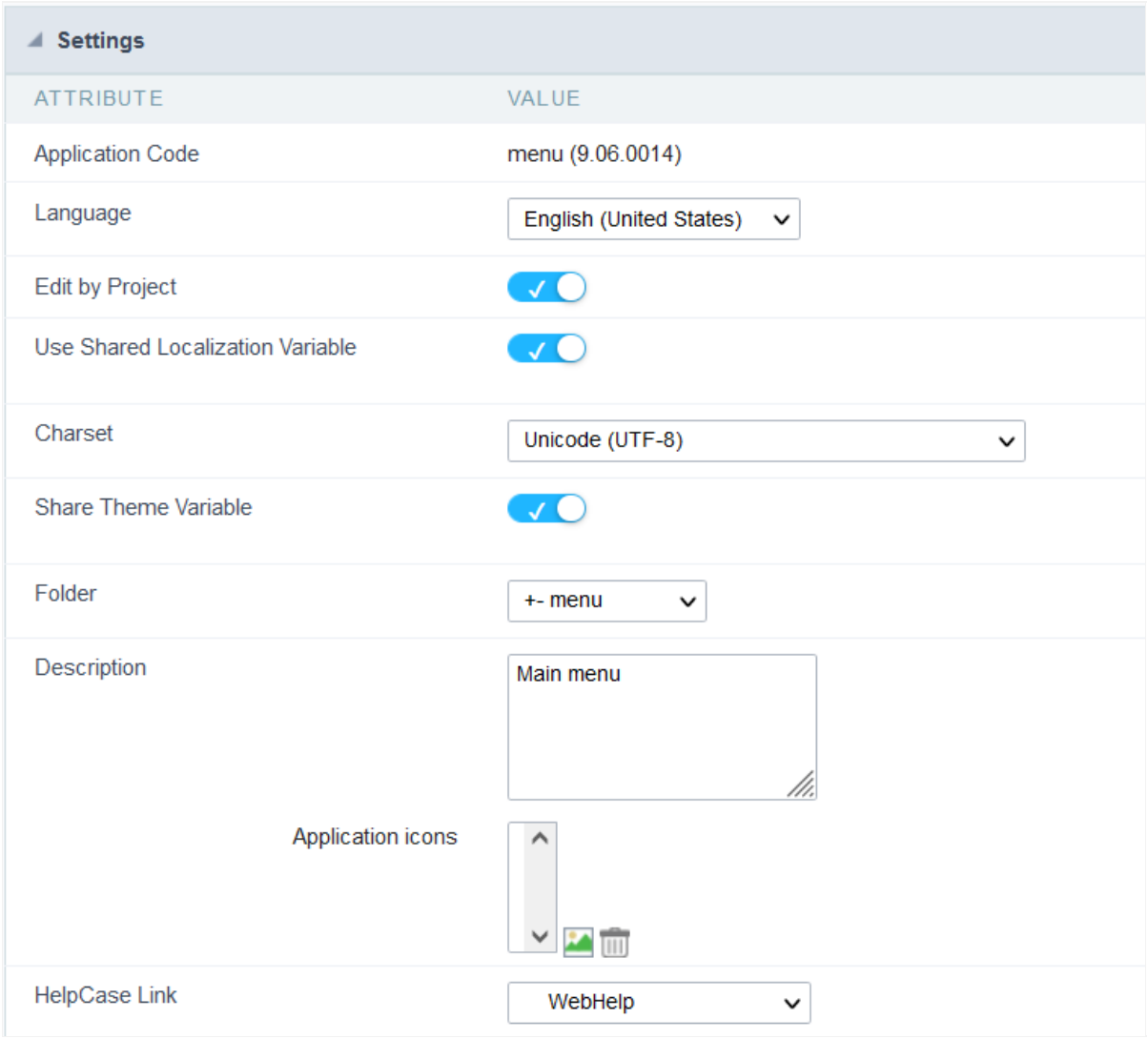
- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Related Samples

Related Videos

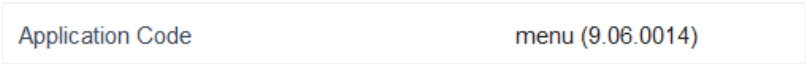
Application Settings (Tree Menu)

Settings



Application Code

Informs the current name of the application and the version in which it was created



Language

Defines a specific language for the application, overriding the default language defined in the project. If no language is selected, the application will inherit the project's default language.

The languages listed in the option are defined in the project properties

Editing by Project

By disabling this option, only the user who created the application will have access to its settings.

No other project users will be able to access the application.

In the image below, the menu application has the edit by project option disabled.

Session Location

Defines if the language of the applications will be stored

Charset

Defines an application-specific charset.

If no value is selected, the application inherits the project's default charset.

Ideally, this charset option, within the application's settings, does not have a defined value. Different usage of your base charset can cause problems saving information in your tables.

Schemes in Session

When active, this option allows the theme stored in the session to be used by the application.

For example, by selecting theme x at login, all applications in the project will use the selected theme.

If the option is unchecked, the application remains with its original theme.

Folder

Folder where the application is stored in Project Explorer.

Description

Displays the description of the application.

You can edit or add a description also in the Project Explorer interface.

Extra Application Icons

It stores the images that will be used in the application through codes in the events, causing these images to be sent along with the application at the time of publication.

Only images used in codes, such as creating a dynamic menu with icons or creating a bill of exchange using a blank, need to be added.

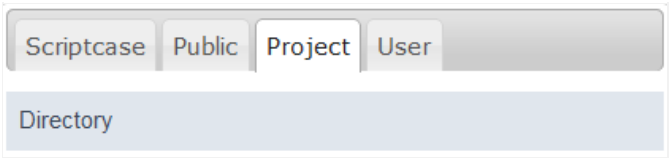
Images added in HTML Image fields or in the application header, for example, are already sent with the applications.

The added images will be stored in this directory: **../_lib/img/**

How to use the images

When inserting the image in the image manager, its name is changed according to the **scope** (public, project or user) and the **storage directory** (background, button, icon, menu or general) following the following pattern.

Scope: It refers to the level of access to the file within Scriptcase, in the development environment.

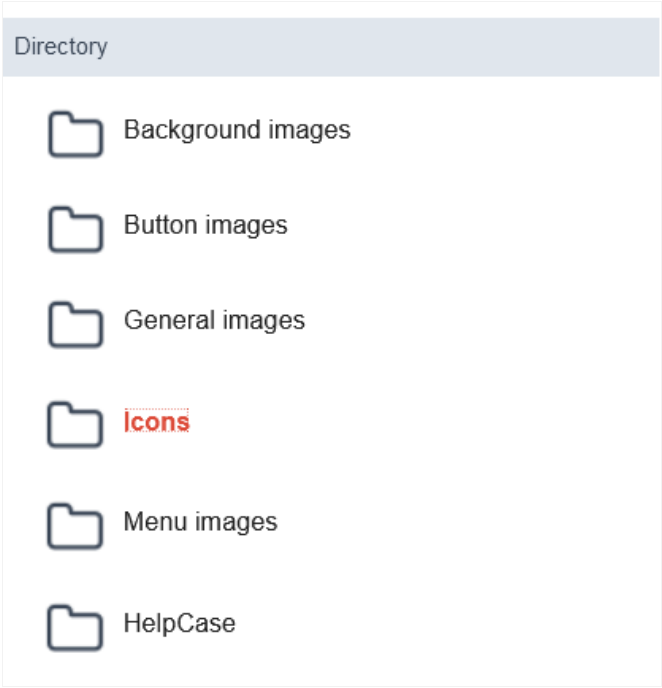


- Scriptcase__ - Contains preexisting images in the tool, where it is not possible to upload new images.
- Public__ - Images accessible for all projects.
- Project__ - Images accessible only in the project in which it was inserted.
- User__ - Images accessible only to the user who uploaded the image.

Each scope adds a prefix to the image name.

- Public - **sys__NM__**
- Project - **grp__NM__**
- User - **usr__NM__**

Storage Directory: Directory where the image will be added. Each directory adds one more prefix to the image name.



- background images - **bg__NM__**
- button images - **btn__NM__**
- general images - **img__NM__**
- icons - **ico__NM__**
- menu images - **menu_img__NM__**

This way, an image called _001.jpg, added in the public scope and in the icon directory will be named **sys__NM__ico__NM__001.jpg**

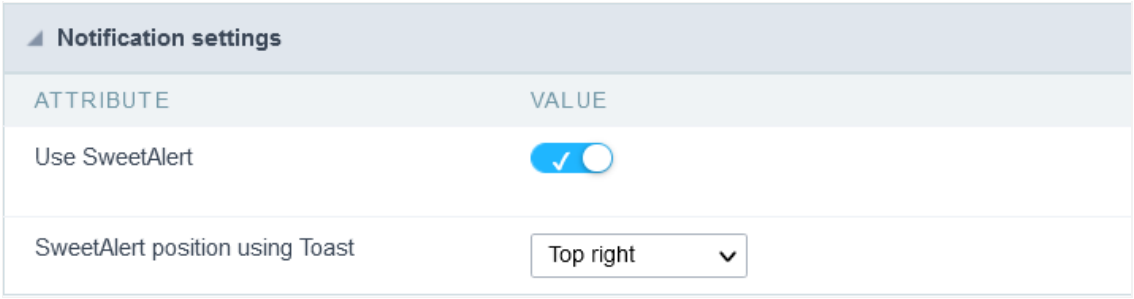
HelpCase Link

Associates HelpCase files to your application.

This configuration can be done expressly in the HelpCase settings, where it is possible to change this option for all project applications. See how by clicking here.

The HelpCase button must be selected in the toolbar to be able to access the selected page.

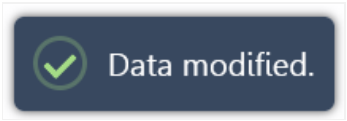
Notifications Settings



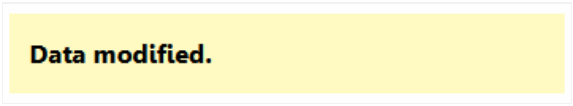
Use SweetAlert

Activate SweetAlert in the application, replacing the “confirm” and “alert” of the browser.

Confirm Example with SweetAlert Enabled

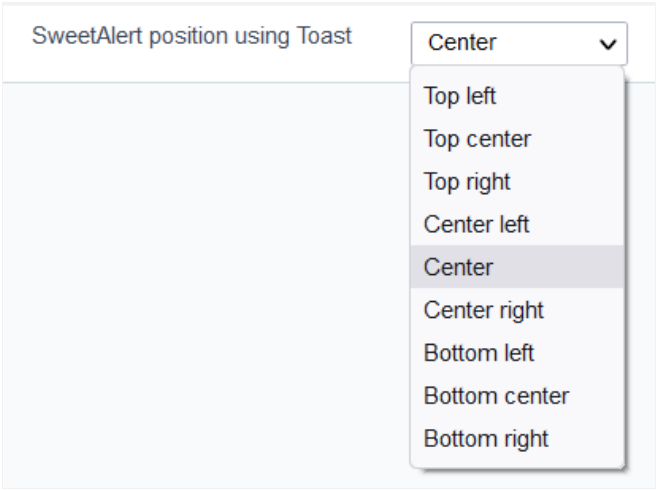


Confirm Example with SweetAlert Disabled

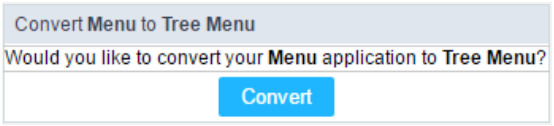


SweetAlert position using Toast

Sets the display position of the message window using toast.



Convert menu type



Scriptcase makes available two types of menu application, they are: Tree Menu and Menu(Horizontal).

This tool allows the conversion of an application from **Menu** to **Menu Tree** or from **Menu Tree** to **Menu**, keeping header and item settings as well as event codes.

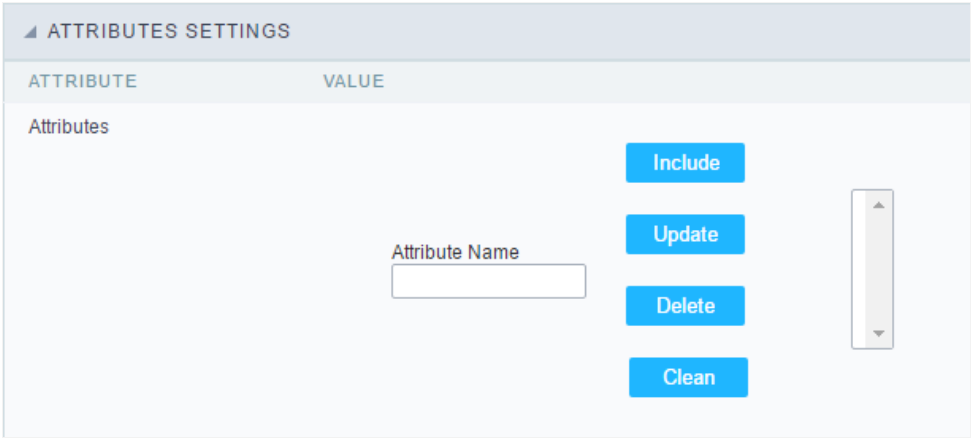
The menu conversion only changes its format, keeping all the settings of the original application.

Tree Menu Programming

ScriptCase has incorporated the concept of Object Oriented programming, using attributes, resources, methods and libraries. It is possible to create your own business rules in applications, and by using these concepts you can reap huge rewards in terms of better organization and improved development.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

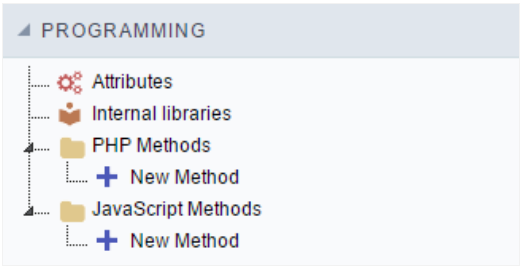
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

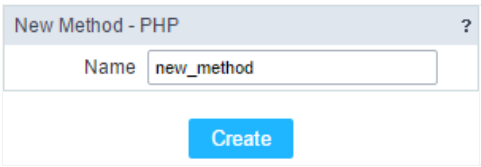
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

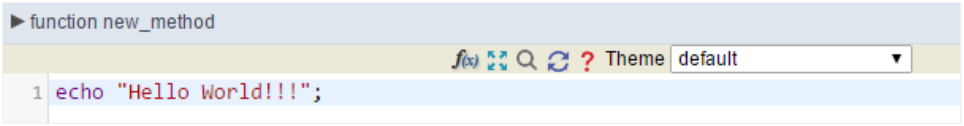


- Give the method a name and click Create. As in the image below.

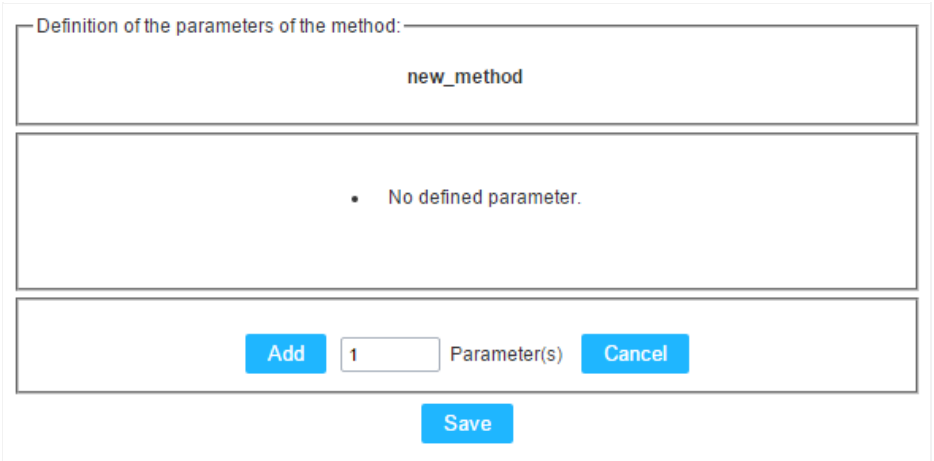
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



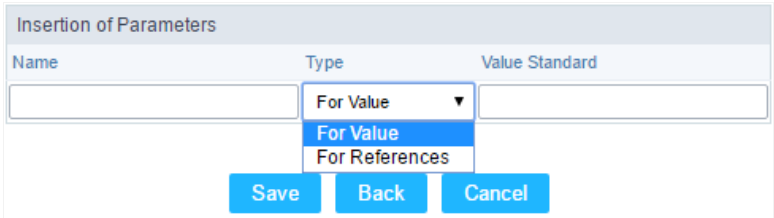
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

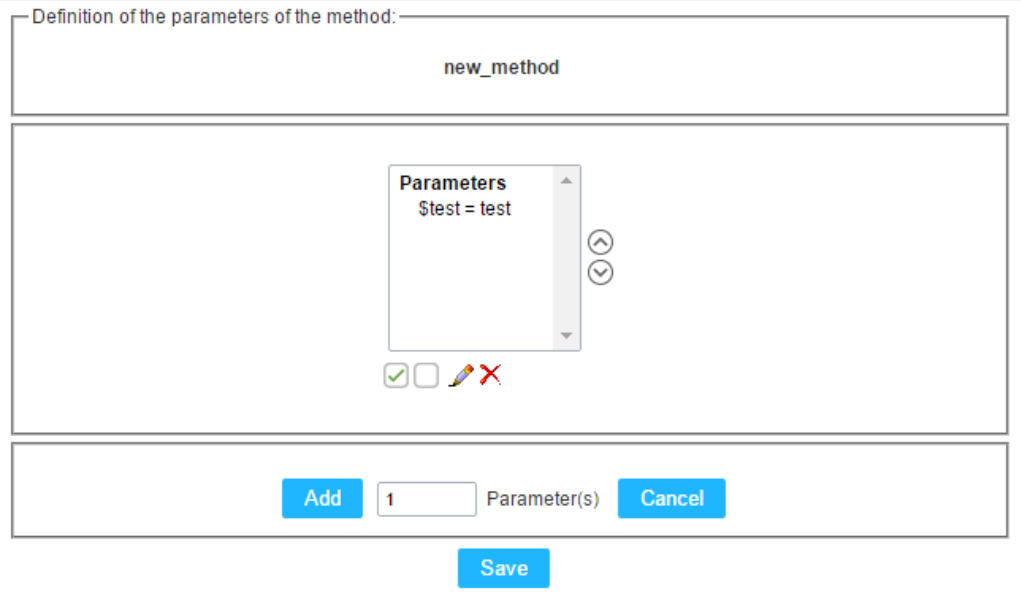


- **Name** : Type in the variable's name.
- **Type** : Selecting the type of variables: For Value or For Reference.

- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:



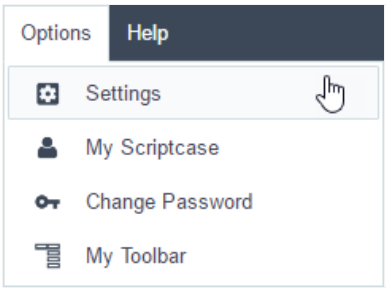
- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

Creating a Tree Menu Application

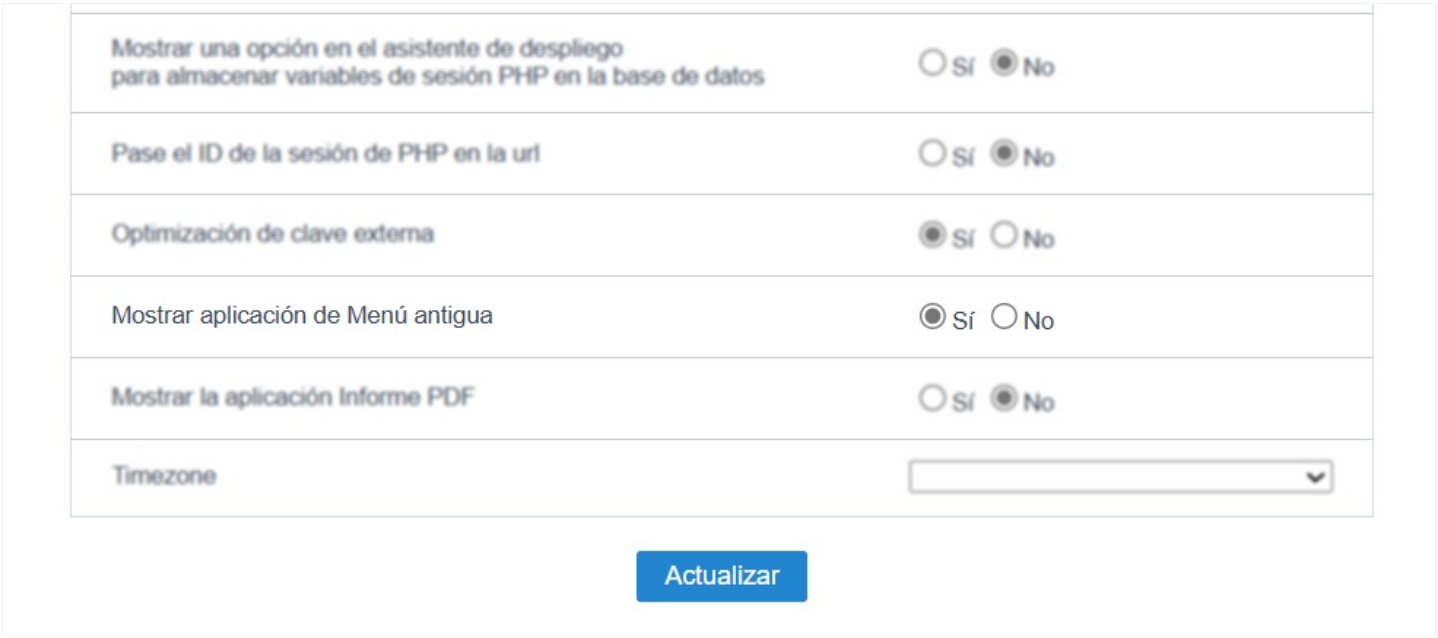
New Application

With the implementation of the responsive menu, the old menu application was removed from the interface. However, it is still possible to use it, keeping in mind that it does not include some features developed exclusively for the responsive menu.

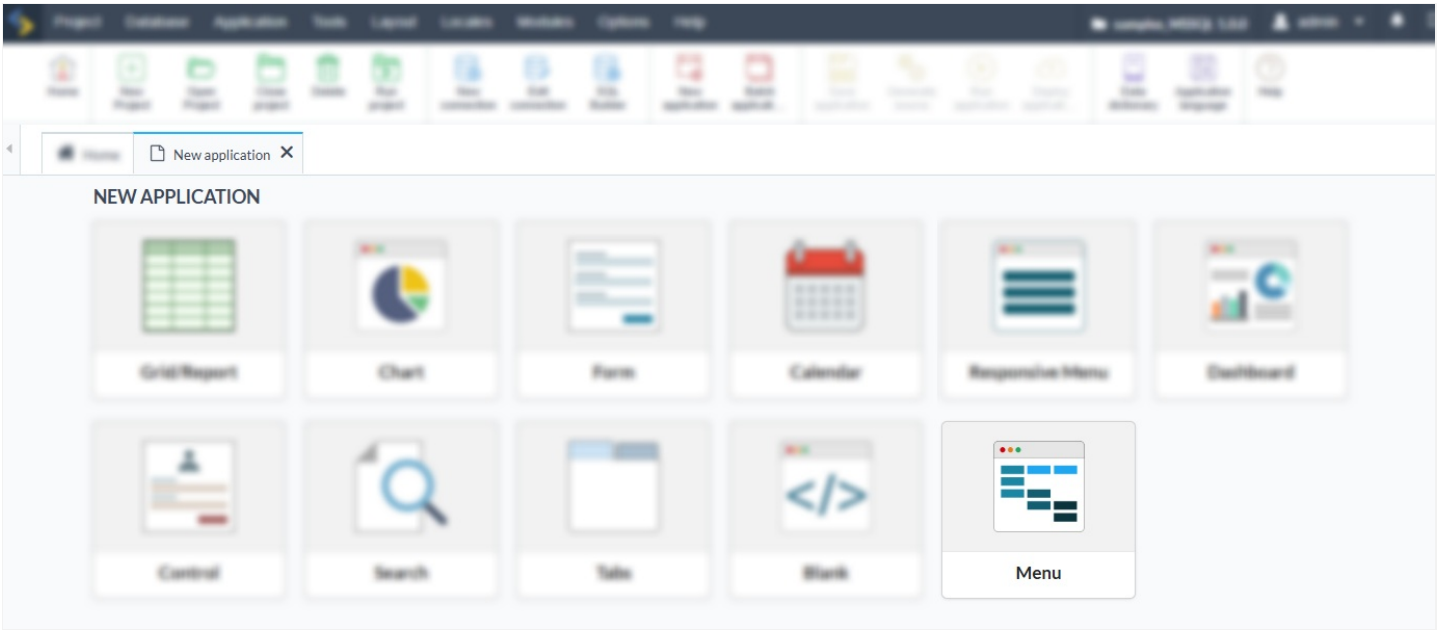
To enable this functionality, the Scriptcase administrator must access the Options > Settings menu and open the System Settings page.



At the bottom of the page, after the foreign key configuration section, you can enable the old menu application to display it as an option when creating new applications.

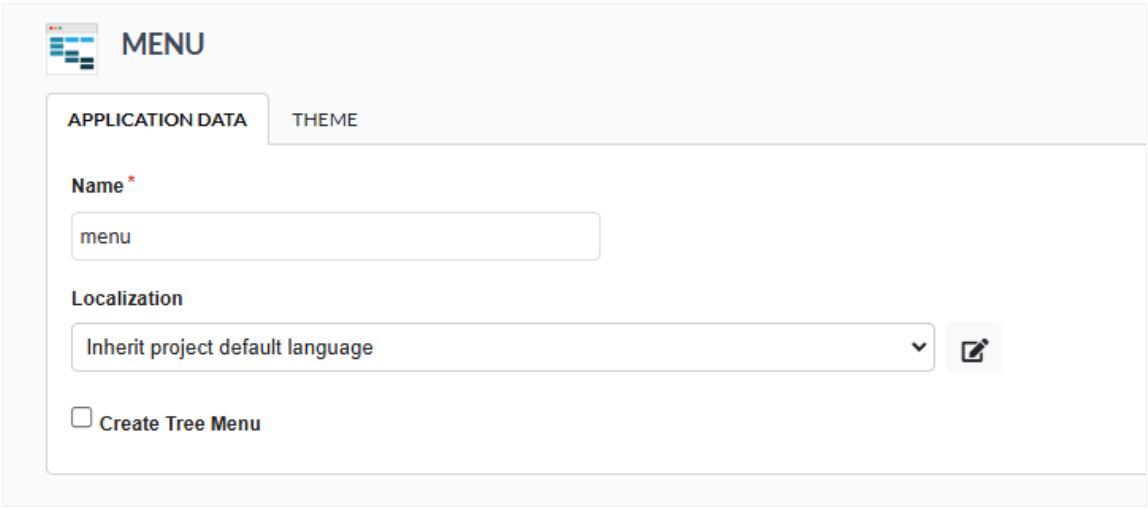


Once enabled, the application will appear on the new application creation screen, as shown in the image below.



Application Data

To create a menu in tree format, simply check the “Create Tree Menu” option when setting up the new menu application. This will automatically enable the hierarchical structure with expandable options.



Name

The name of the application being created. It cannot contain special characters.

Localization

The language of the application to be created. The project’s default language is automatically selected.

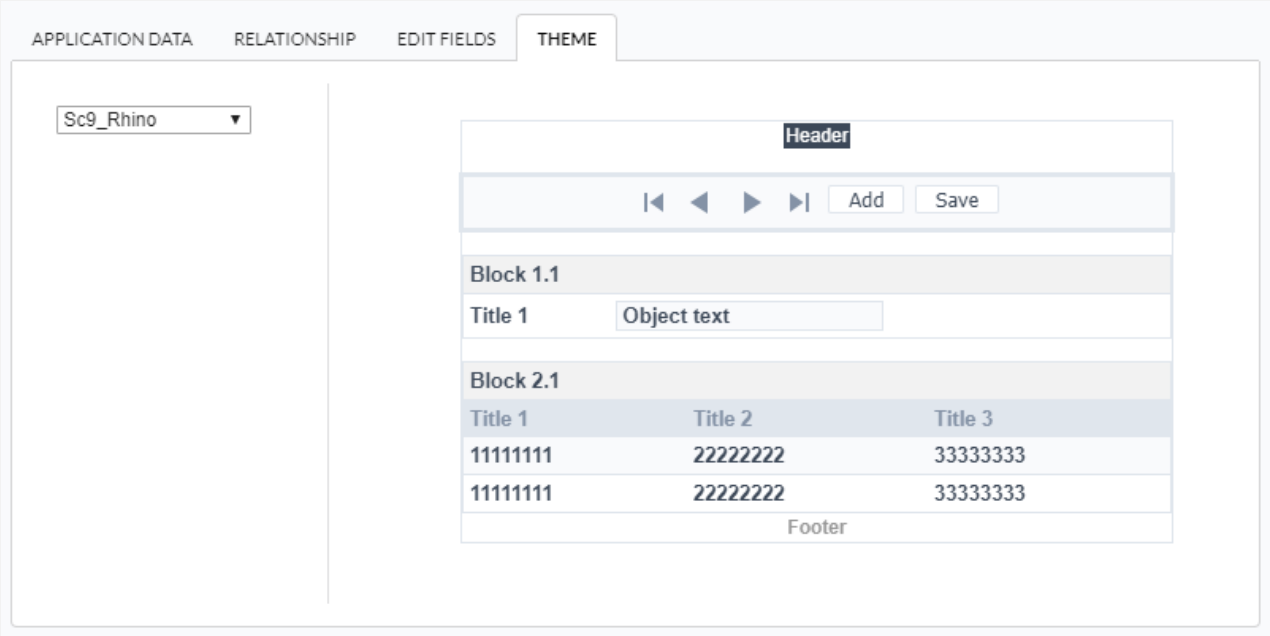
Create Tree Menu

Allows you to create a menu application in a tree format, where the elements are organized hierarchically with expandable options.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and

modified in **Project** > **Properties**, is automatically selected.



Related Samples 🔗

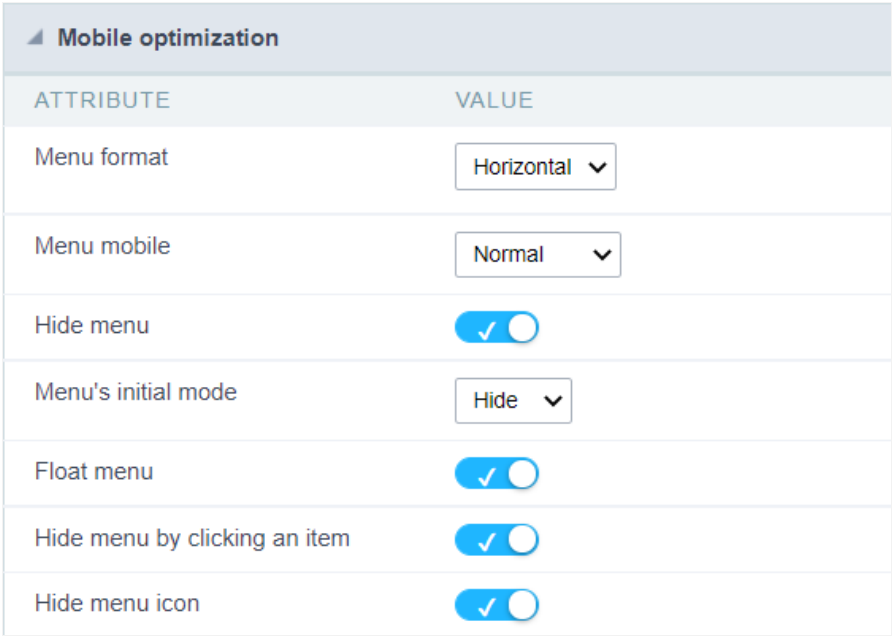
Related Videos ▷

Tree Menu Mobile Optimization

The Mobile settings allows to define the automatic optimization of generated applications to run on mobile devices.

See below the available settings.

Mobile optimization

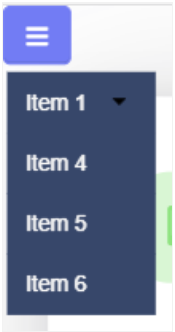


Menu Format

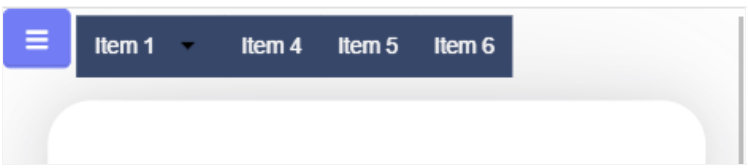
Sets the menu items orientation when the **Menu mobile** option is set to **normal** value

The menu can be displayed vertically or horizontally. See the examples below.

Vertical menu example



Horizontal menu example



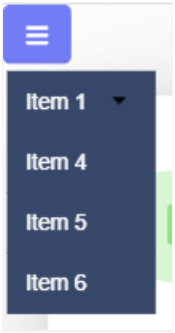
Mobile Menu

Sets the menu display format in two ways: **normal** and **navigation**

Normal

Sets the menu display format in two ways: **normal** and **navigation**

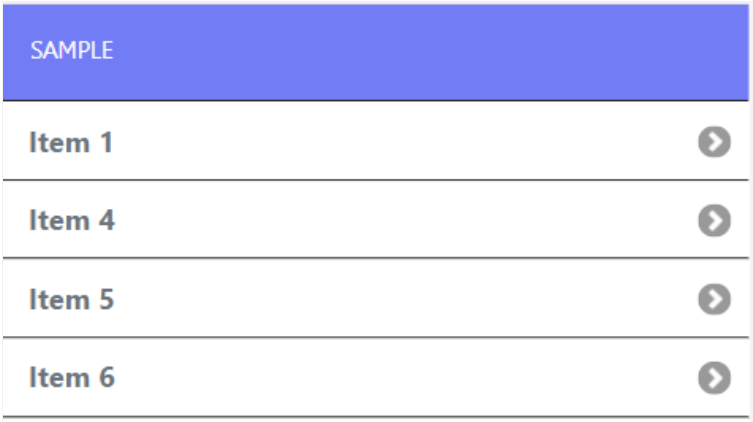
Sample normal menu



Navigation

This type of menu expands and takes up the entire screen, so that navigation is limited to only the options of the selected item.

Sample Navigation Menu



hide the menu

Defines whether the menu can be collapsed or will remain fixed on the screen.

When **enabled**, this feature allows the end user to hide the menu. If the developer chooses to **disable this option**, the menu items will continue to be displayed on the screen.

The behavior of the hide menu option is defined in the detailed options below and are enabled only when activating the hide menu feature.

Initial state of the menu

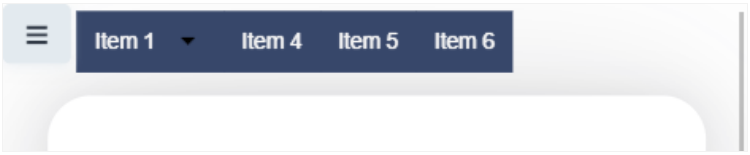
Only available when the **Hide menu** option is enabled. This setting defines the initial state of the menu.

The options are **Open** and **Hide**.

Open

This value displays the menu items on the initial access to the application, without the need to click on the button to expand the menu.

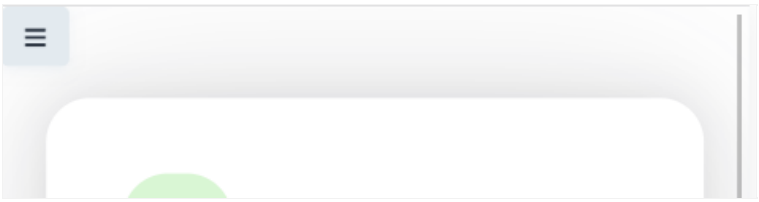
Example of open starting menu



Hide

In this option the menu items will not be displayed in the initial load of the application, it is necessary to create the button to expand the menu to perform the navigation.

Example of menu starting closed



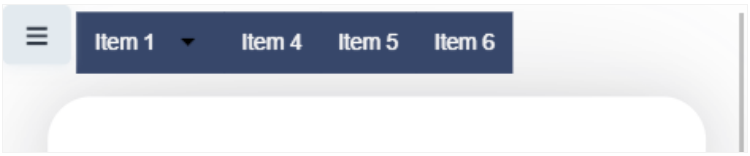
float menu

Defines how the items are opened in relation to the application.

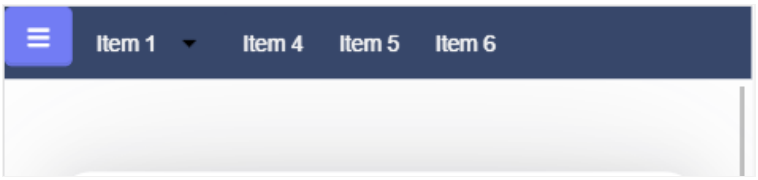
When enabled, items will be displayed floating, overlaying the application without changing its display.

When disabled, items will be opened together with the application, occupying space and moving it away during the period it is open.

Example with option enabled



Example with the option disabled



Hide menu when clicking an item

This feature allows menu items to be collapsed when clicking on one of the items to open an application.

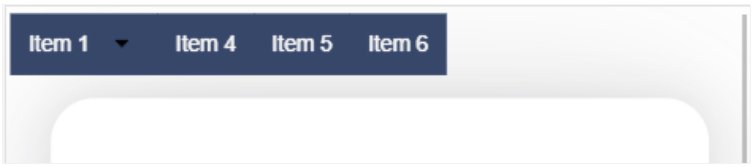
Hide menu icon

Defines whether the Hide Menu button will always be visible.

This option is only available if the **Hide menu when clicking an item** is enabled.

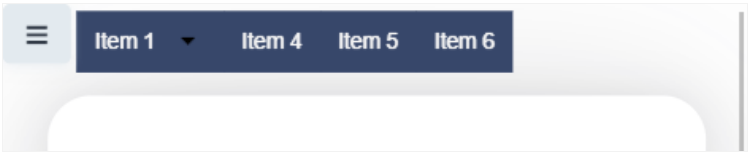
Enabled

In this case, the hide menu button will also be hidden when clicked, returned to display as soon as an item is clicked to open an application.



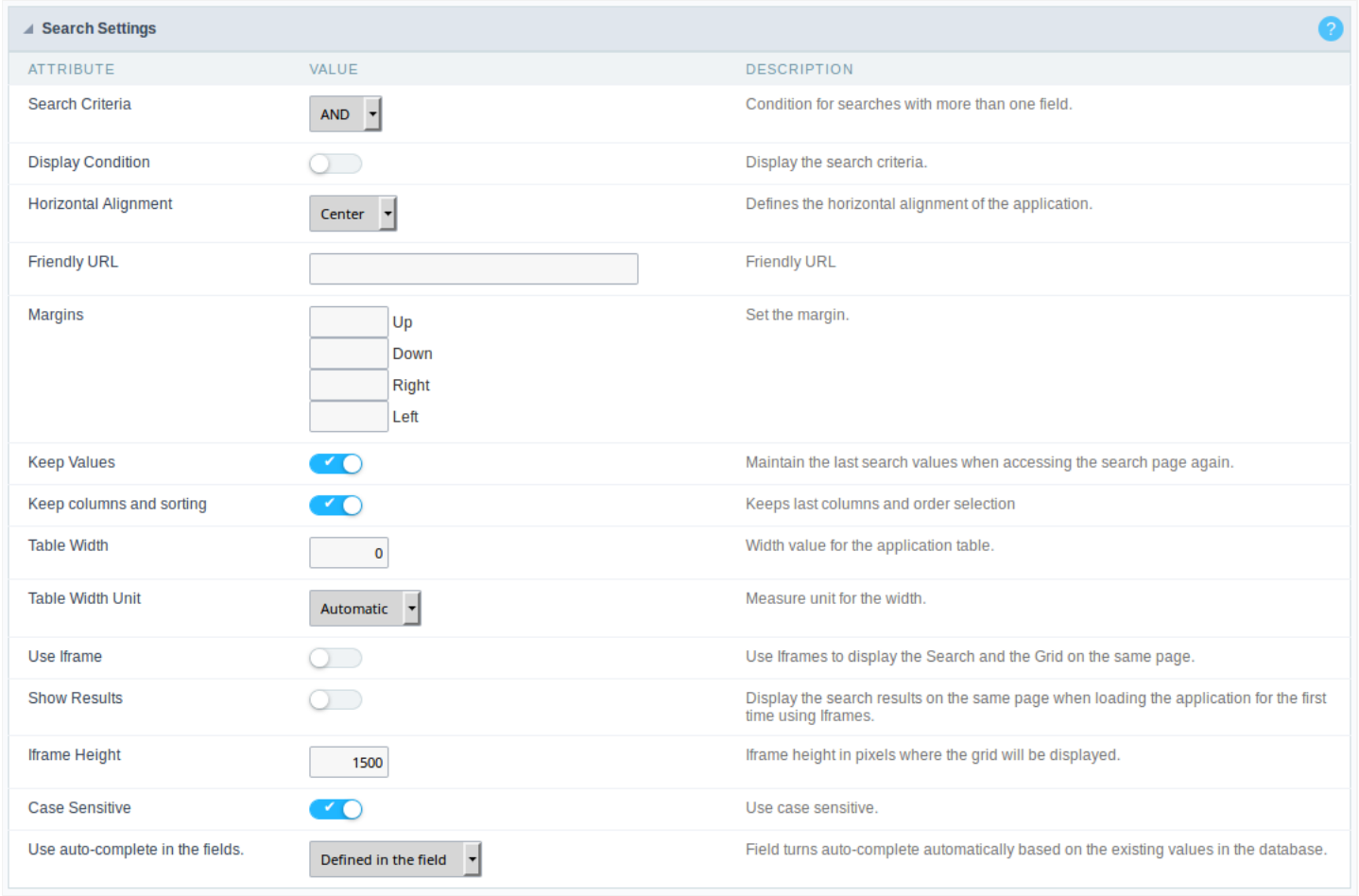
Disabled

In this case, the hide menu button will always be visible, even when items are being displayed.



Search Application Settings

Through these attributes you can define some general search settings such as alignment, width, search conditions, etc.



Search settings interface.

- **Search Criteria** : Using this option you can select the SQL command as “AND” or “OR” to set the search criteria to be used;
- **Display Condition** : If you choose “Yes”, it will be displayed all records when the search fields are blank. If the option set is “No”, no record will be displayed;
- **Horizontal Alignment**: It allows to define the horizontal alignment of the application (Centered, Left, Right).
- **Vertical Alignment**: Allows you to define the initial vertical alignment of the application (Above, Centered and Below).
- **Friendly URL** : This field allows you to change the URL that will be called by the application. Allowed characters are the same available on URLs: a-z, A-Z, 0-9, -_. This option can also be changed on the home screen, at the “Friendly URL” column at the applications list;
- **Margins** : Allows you to set the Search application positioning (margins).
- **Keep Values** : When this option is activated the last searched values are going to be maintained and displayed when the system user returns to the search application screen.
- **Keep columns and sorting** : This option determines if each search should preserve the selected values to the Grid fields columns and sorting, i.e. to each search, these values return to original condition.
- **Table Width** : This option sets the search application HTML width value, it can be in pixels or percent. You need to set the unit within the option “Table Width Unit”.
- **Table Width Unit** : Using this option you can set the Search application width unit;
- **Use Iframe** : When this option is enabled the Search screen and the search results are going to be displayed in the same browser window, using two FRAMES, one above the other. The Search application and the Grid with the results are going to be part of the same page.
- **Show Results** : This option is associated with the “Use Iframe”. When enabled it will initially displays the Search application along with the Grid inside the iframe positioned below the search screen, when disabled initially displays only

the Search screen.

- **Iframe Height** : This option is also associated with the “Use Iframe”. It sets the height, in pixels, of the iframe where the recovered data will be displayed by the search.
- **Case Sensitive** : This option enables the Case Sensitive option.
- **Use auto-complete in the fields** : Using this option you can enable the auto-complete option automatically based on the existing values in the database tables. You can also define this option one by one using the “defined in the field”

Related Links 🔗

Related Video ▷

Advanced search

Select fields

Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

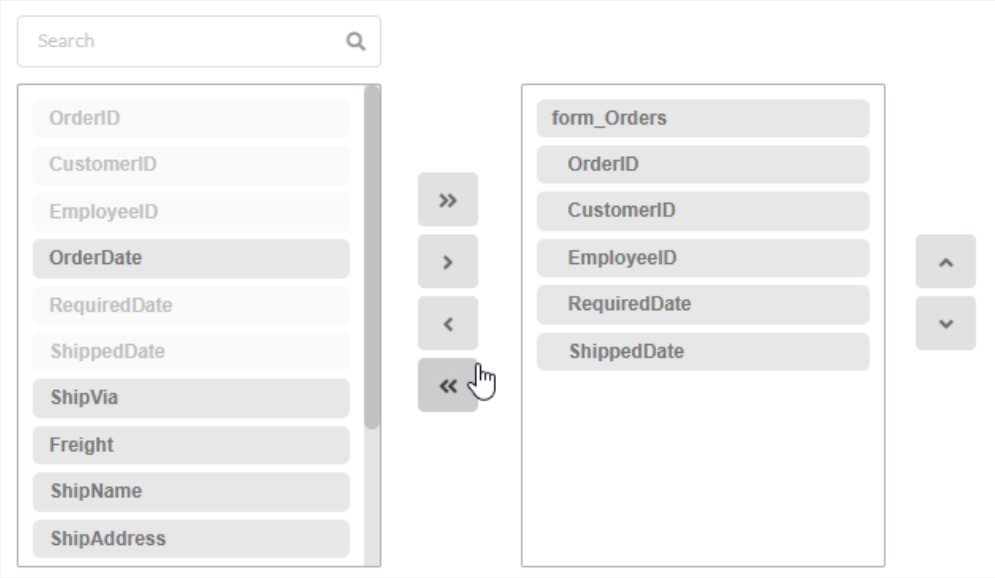
- » - Move all fields to the right.
- › - Moves only selected fields to the right
- ‹ - Moves only the selected fields to the left.
- « - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the › button.

To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning



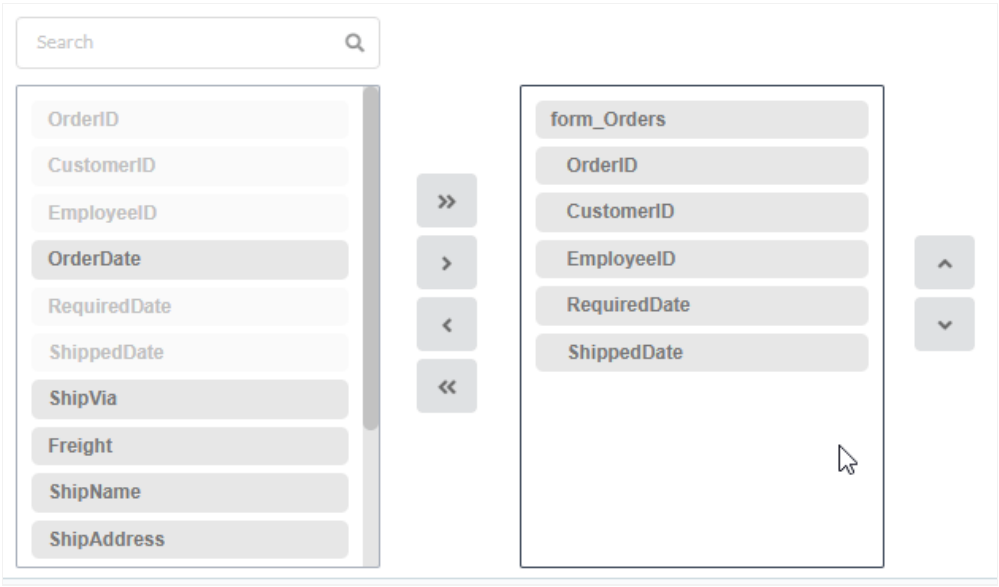
Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

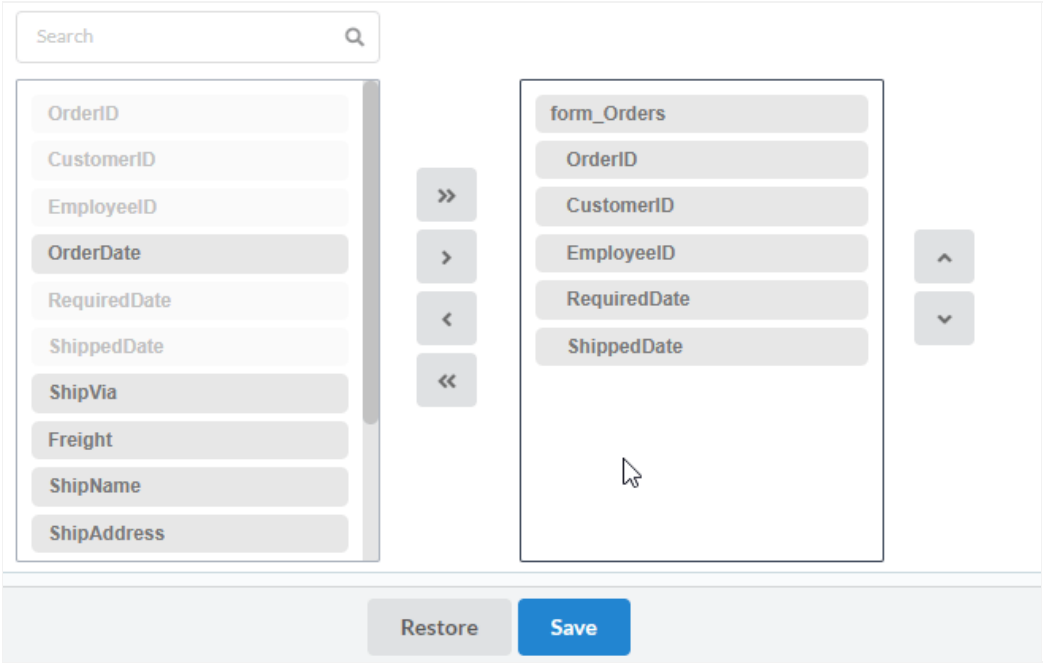


Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

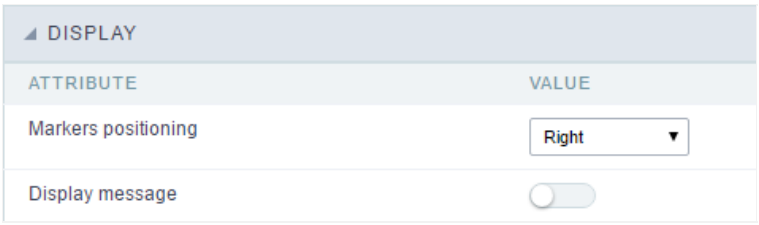


Required

This interface has the options for you to select and set the Search application required fields.

Required fields interface.

Within the generated application will be displayed a marker(*) next to the field and an error message will also pops up if there's no value assigned to the mandatory fields. You will be able to set some options for the market using the options below.

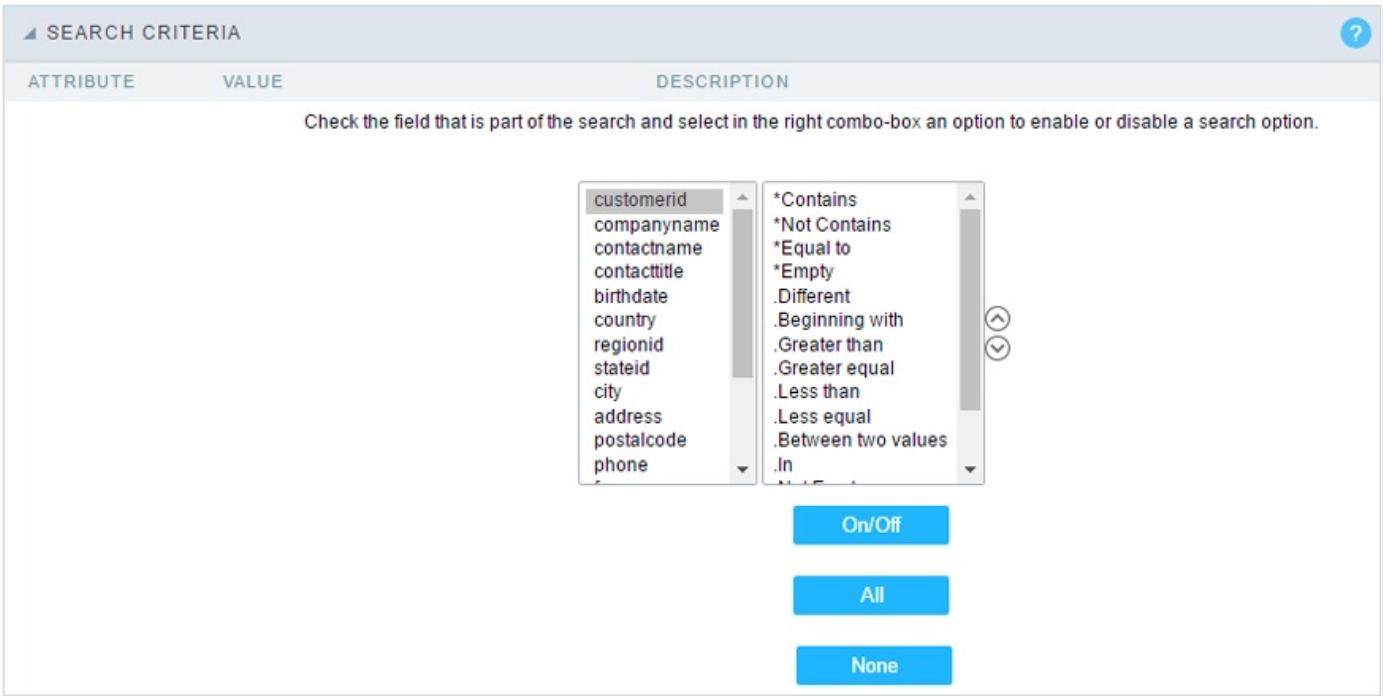


Marker placement configuration interface.

- **Marker position** : Marker's position relative to the field (options are right, left or Do not display).
- **Display message** : Displays whether or not the validation error message.

Search Criteria

With this interface, you can configure the conditions available for each field of the Search form.



Search configuration Interface.

We can see the fields list on the left combo. On the right, the list of options for filtering the selected field. To select an option, click on one of them (Equal to, Beginning with, Contains, etc.) and then the button On/Off. The arrows, on the right, allows altering the order of the fields.

For the Date type fields, you can define special conditions for the search, accessing the field configurations, and editing the Special Conditions Settings.

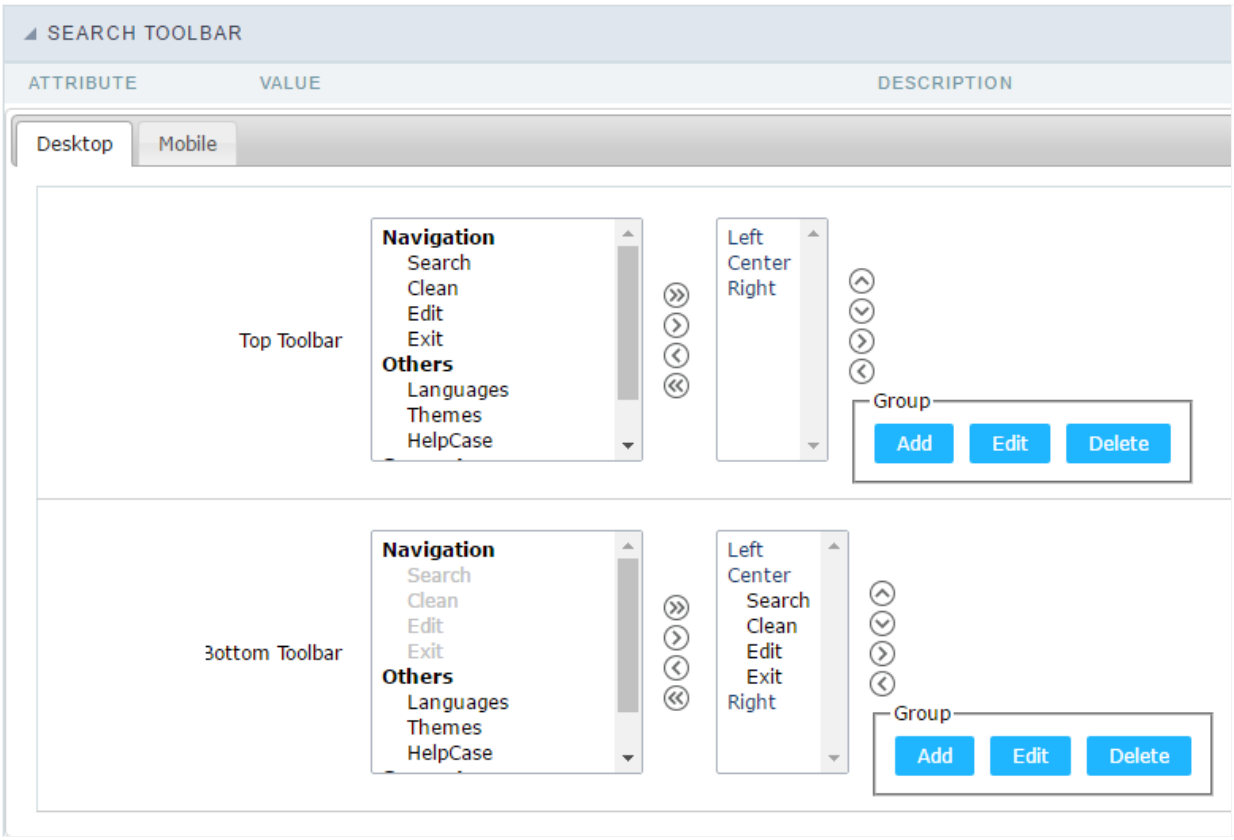
Below the list are the buttons to enable the selected options:

- **On/Off**: Enables or disables the field or the option chosen.
- **All**: Marks all fields or options.
- **None**: Unmarks all the fields or options.

Toolbar

Search Toolbar

The Search toolbar is divided in two parts: Top and Bottom, in a way that is possible to define the buttons that will be displayed in both bars. The selection of buttons in the top and bottom toolbar works independently, allowing the buttons to be displayed in both bars at the same time.



Toolbar Interface.

Navigation:

Groups the options relative to the navigation buttons that can be displayed in the application.

- ***Search***: Execute the search.
- ***Clean***: Clean the all the search fields.
- ***Edit***: Enable the ***Save Tag*** option.
- ***Exit***: Exit the application.

Others:

Groups a diversity of options relative to the application.

- ***Languages***: Displays a combobox with the names available, defined in the project properties.
- ***Themes***: Displays a combobox with the themes available, defined in the project properties.

- ***HelpCase***: Displays a button to redirect to the help page.

Separator

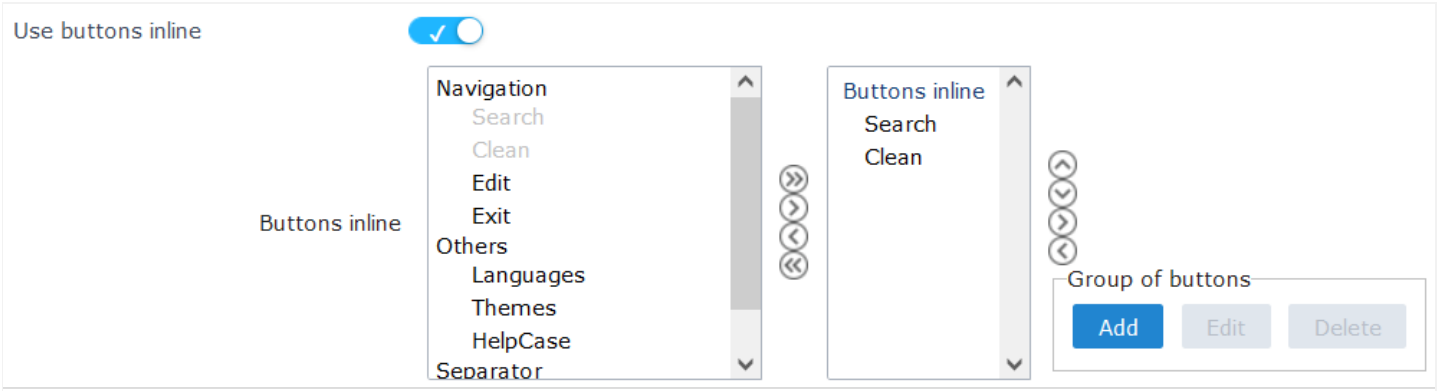
- **−−−−−−**: Dispays a line separating the buttons, when used the Group Buttons.

Use in-line buttons:

Allows the alignment of the filter buttons next to the fields.

Inline buttons: Allows you to select which buttons will be displayed next to the field, and you can sort them according to your wishes. This option is available by enabling **Use buttons inline** in the button settings.

When activating the **Inline Buttons** option, the screen to configure the buttons will be displayed.



The buttons available in this in-line button option are the same as those shown in the standard toolbar described above.

This is the result when using a radio button on the line

Button Settings

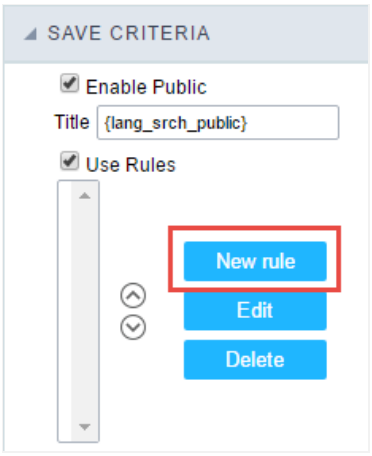
Button Settings Interface.

- **Hotkey**: Allows you to set keyboard hotkeys to a button.
- **Position of the in-line buttons**: Sets the positioning of the buttons to the right or left of the fields.
- **Column Quantity**: Sets the number of display columns of the buttons, allowing you to configure whether they will be displayed side-by-side or distributed in columns.

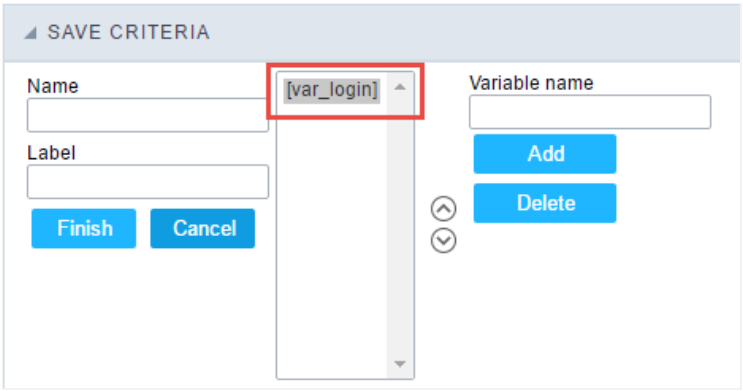
Options ![Options Interface.][barra_ferramentas_filtro_opcoes] *Options Interface.* * __Button Position(Top/Bottom)__ Positioning the buttons of the toolbar Top/Bottom. -->

Save Search

This feature allows the end-user to save his searches in a profile. You can create some rules, like to save the searches by user login.



Save Filter Interface.

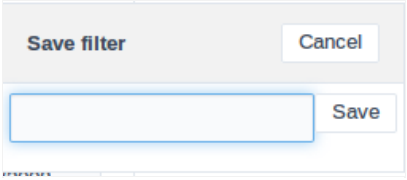


Save Filter Interface.

Save Tag

This feature allows the end-user to save his tags of searches in a profile. To save a search, he must click on the “Edit” button, then add a name to the profile and click on “Save Filter”.

Search Tags.



Save Tag Interface.

Layout Settings

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Blocks

Blocks are “containers” where you can position the application fieldSlides of Forms, Controls, or Grids.

Scriptcase creates applications with one block by default. You can add more blocks as you wish, to organize it in the best way.

See below, the Columns Organization, and where you can define the position of the next block: beside or below the current one.

Application Block configuration

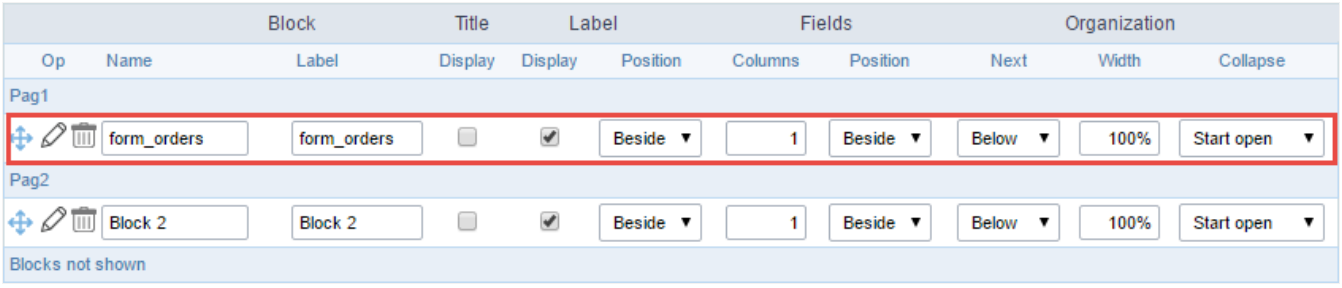
On the left side of each block, there are two icons, the first one to edit the information of the block and the second one

to delete the block.

Organizing the position of the Blocks

See below how to modify the display order of the Blocks in one Page.

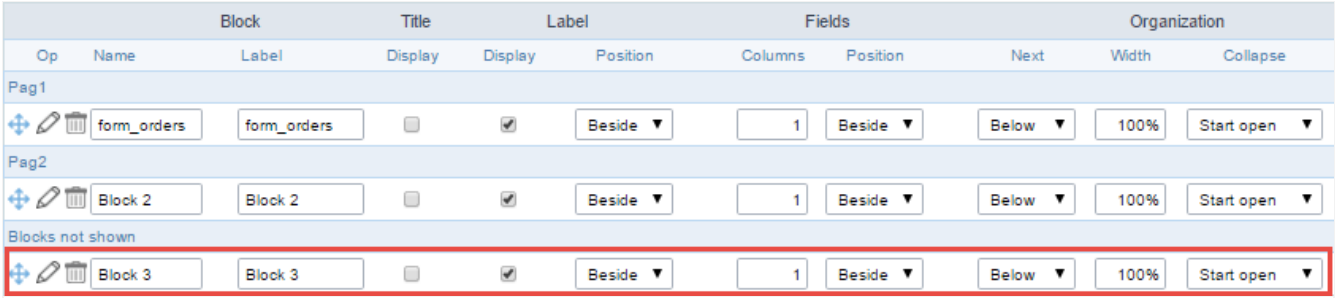
Click and drag the block that you desire to modify to its new position.



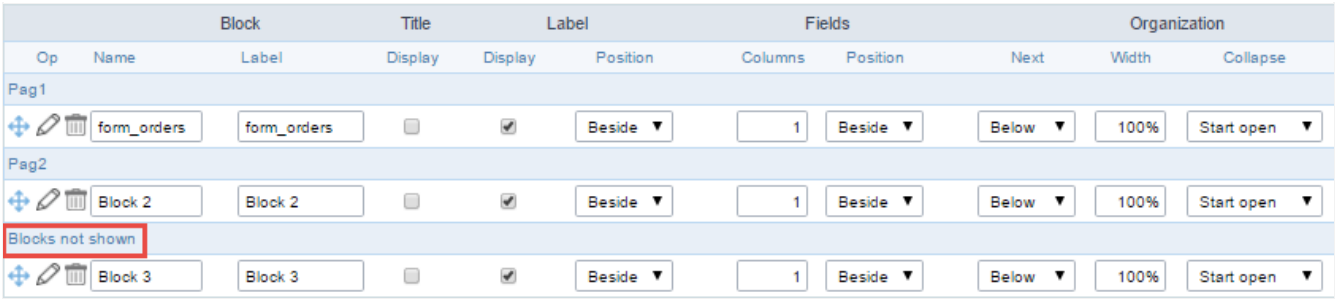
Application Block Display configuration

- See how to remove a block from the display

Click on the block desired and drag it to the item “Blocks not Shown”. This way, you can also drag the block to another page if desired. See the images below.



Application Block Display configuration



Application Block Display configuration

Block

- **Name**: The name that identifier the Block.
- **Label**: Title of the block to display in the application.

Title

- **Display**: It controls the display of the block title.

Label

- **Display**: It controls the display of the field labels of the block.
- **Position**: Options to display label :
    1. **Above**: Display the label above the field.
    2. **Beside**: Display the label beside the field.
    3. **Below**: Display the label below the field.

Fields

- **Columns**: Amount of columns side by side in the block.
- **Position**: The way to display the fields in the block:
    1. **Below**: Display the fields one below the other respecting the number of columns.
    2. **Beside**: Display the fields one beside the other respecting the number of columns.
    3. **Line**: Display the fields one beside the other with no tabulation.

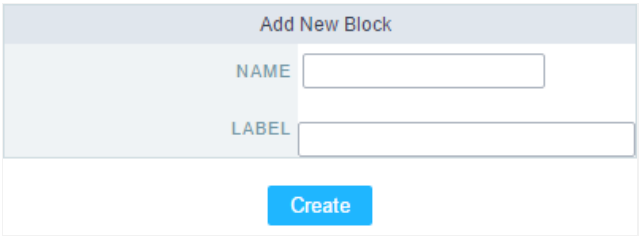
Organization

- **Next**: The way to display the blocks in the page:
    1. **Below**: Set to show the following block below the current one.
    2. **Beside**: Set to show the following block beside the current one.
    3. **Tabs**: Set to show the following block in a different tab then the current one.
- **Width**: Set the block width in pixels or percentages. Use the symbol “%” to indicates the value in percentage.
- **Collapse**: Enables the option to close the block.

Create a New Block

To include new blocks in an Application, click on the button

. Then, enter the name and label of the block in the following interface and finish by click on Create.



Creating application blocks configuration

Name

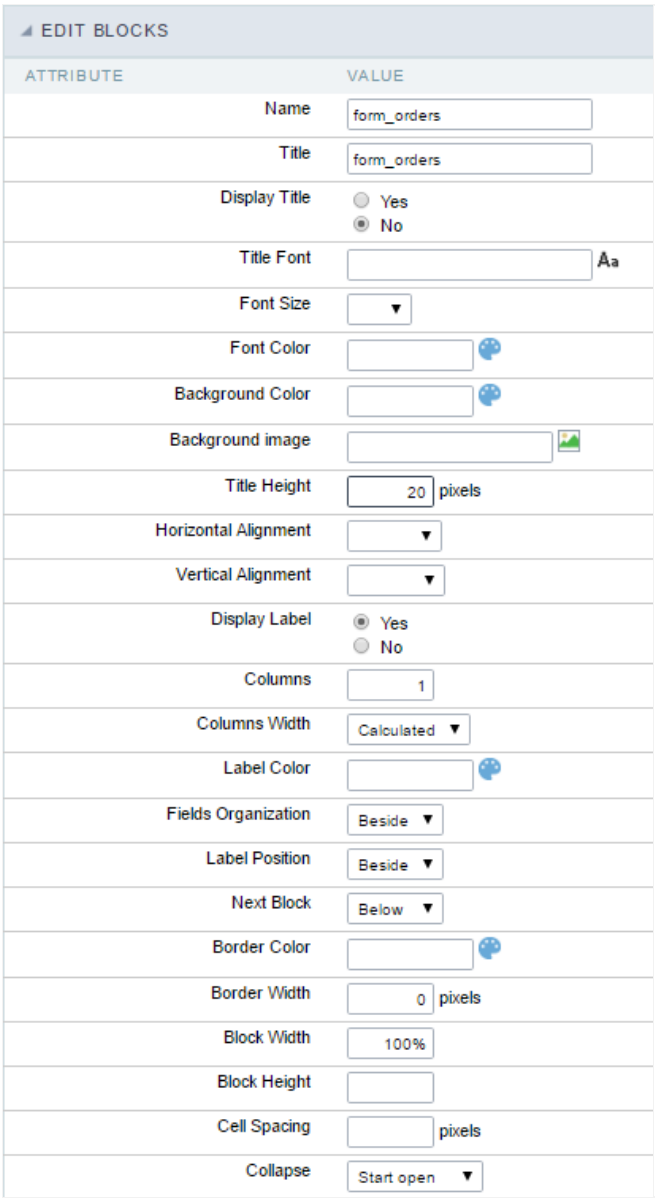
Name of the Block.

Label

Title of the block to display in the application.

Edit Blocks

To edit a block, click on the icon , that is on the left side of the block. Then you can see the following interface to define the parameters of the blocks. Click on Save to finish.



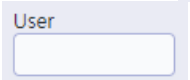
Application Block editing interface

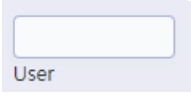
Name

Name of the block. #### Title
Block title for display. #### Display Title
This option, when active, allows displaying the block title. #### Title Font
Set the font family of the block title. #### Font Size
Set the font size of the block title. #### Font Color
Set the font color of the block title. #### Background Color
Set the Background Color of the block title. #### Background image
Set a Background image for the block title. #### Title Height
Height in pixels of the block title line. #### Horizontal Alignment
Horizontal Alignment of the block title (Left, Center, and Right). #### Vertical Alignment
Vertical Alignment of the block title (Top, Middle, and Bottom). #### Display Label
Display the labels of the fields in the block. #### Columns
Amount of field columns in a block. #### Columns Width
Set the field column width of the block. #### Label Color
Color of the field labels. #### Fields Organization
The way to display the fields in the block. #### Label Position
Set the position of the field labels of the block.

The options are:

- **Beside** - This option positions the label on the right side of the field.

- **Above** - This option places the label above the field. 

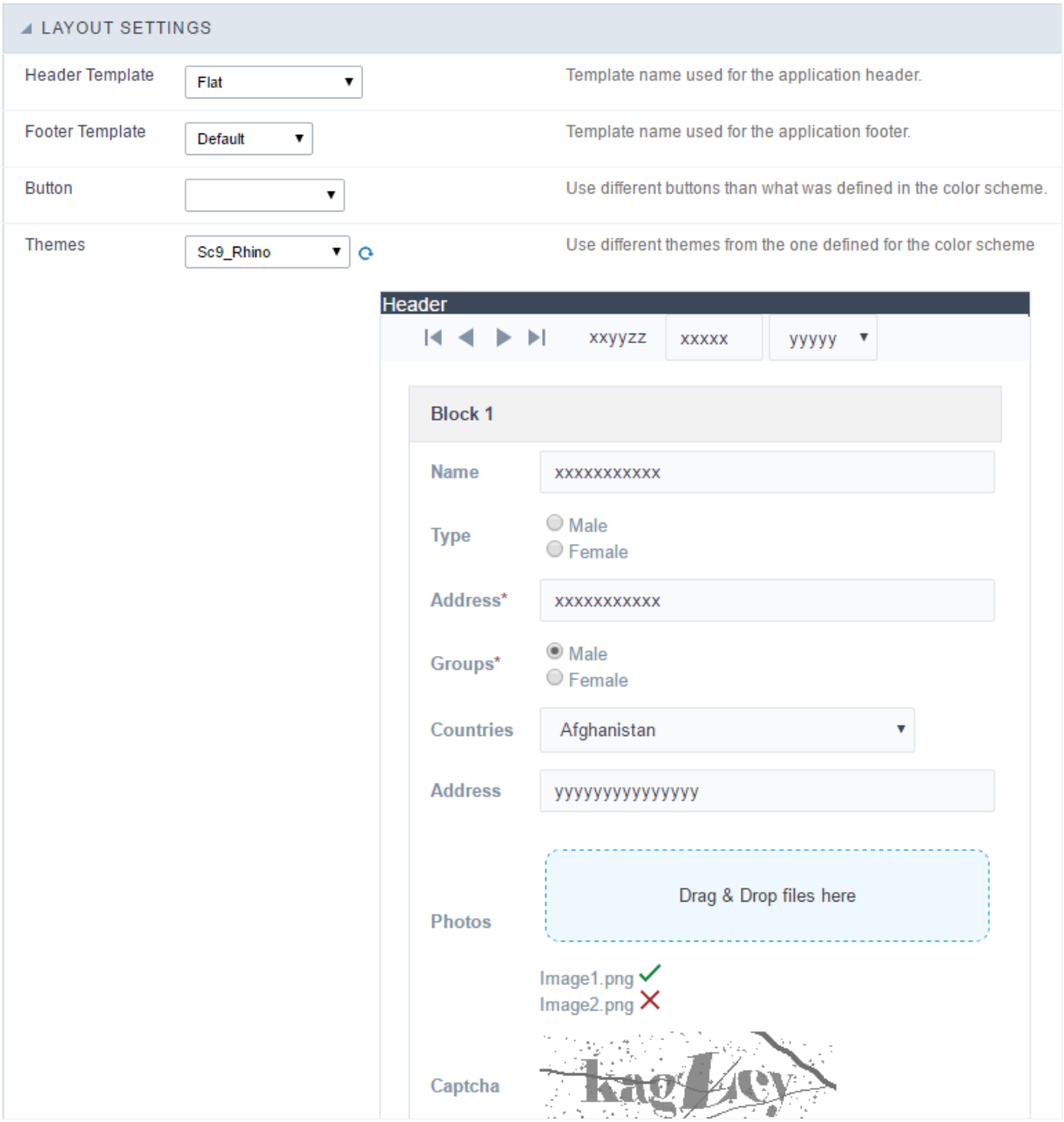
- **Below** - This option places the label below the field. 

Next Block

Set the position of the following block relating to the current one. #### Border Color
The border Color for the block. #### Border Width
The border Width for the block. #### Block Width
The width for the block. #### Block Height
The Height for the block. #### Cell Spacing
The Cell Spacing in the block. #### Collapse
It enables the option to close the block.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout Settings

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

Allows choosing the button theme for the application.

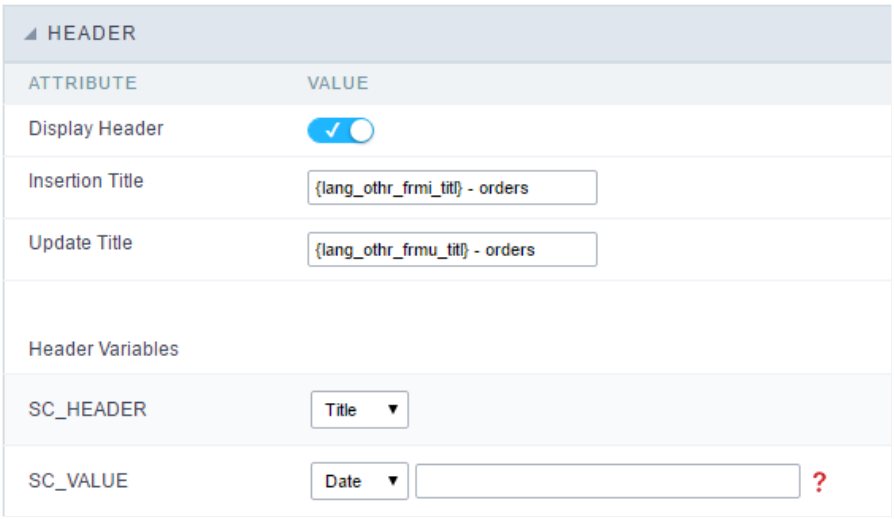
Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks

according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

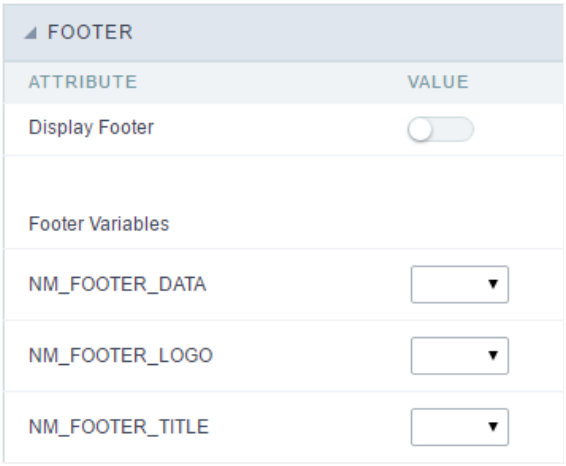
Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

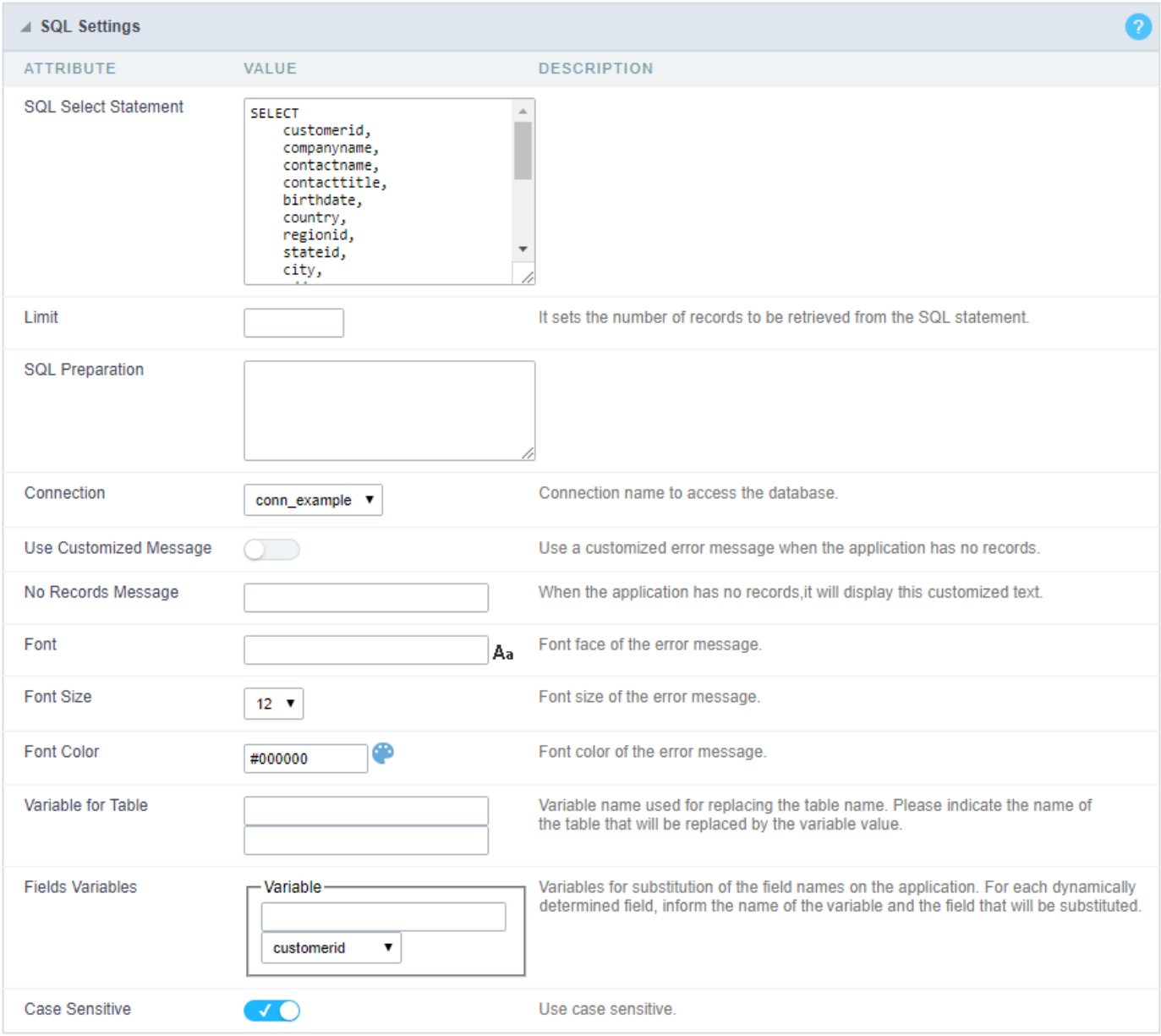
Related Links

Related Video

Search Application SQL Settings

SQL Settings

This interface allows configuring the related database settings, such as the SQL statement, the used database connection, case sensitive, and others.



SQL configuration

SQL Select Statement

It allows you to define the primary SQL of the application. You can edit this SQL to add or delete fields.

Limit

Lets you limit the display in the number of records retrieved by SQL query.

SQL Preparation

You can enter SQL commands or procedure names to execute them before the primary SQL of the application.

Grid

Connection

It allows defining the database connection of the application. You can change the connection to another one that has the same table.

Use Customized Message

Lets you define to display the “no records” message or not.

No Records Message

Lets you set the message when the application has no records.

Font

This option is available when using the “Use Customized Message” option. It lets you set the font for the message.

Font Size

This option is available when using the “Use Customized Message” option. It lets you to set the font size.

Font Color

This option is available when using the “Use Customized Message” option. It lets you to set the font color.

Variable for Table

It allows to use a variable to change a part of the string containing the table name.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the table you want to replace (replace from).

Fields Variables

It allows to use a variable to change a part of the string containing the field name.

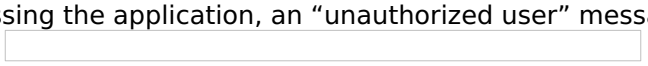
Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the field you want to replace (replace from).

Case sensitive

It defines if the database connection uses case sensitive or not.

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application. 

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Headers

Disable XSS Auditor

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Captcha

Captcha

Use Captcha

Defines if the application uses Captcha.

Number of Characters

Amount of characters in the Captcha image.

Character List

List of characters used in the Captcha.

Label

The message displayed for the Captcha.

Error message

Captcha error message.

Height

Height of the Captha image. (in pixels)

Width

Width of the Captha image. (in pixels)

Font Size

Font Size of the Captha image. (in pixels)

Reload

Display the refresh button in the Captcha.

Select one layout

It offers layouts for display the Captcha.

Recaptcha

ReCAPTCHA is an API provided by Google for forms. It adds security, preventing automatic submission of forms through robots.

reCAPTCHA sample:

1. First, we must request an API Key to activate reCAPTCHA into a Scriptcase application by following the steps below:

   To get a **Site key** and **Secret Key** go to the link: https://www.google.com/recaptcha/admin#list.
   See the image:

Label

It is a project identifier to create the reCAPTCHA keys.

Choose the type of reCaptcha

We must choose the option **reCAPTCHA V2**.

Domains

We can insert multiple domains (one per line) to limit the API uses.

1. Then, we need to accept the Terms of Service (“Accept the reCAPTCHA Terms of Service”).
2. When clicking on **Register**, the page refreshes and shows the integration of reCAPTCHA information. There we can get the **Site Key** and **Secret Key**:

1. Now, we can set the Scriptcase application security:

Site Key

Key generated by Google after reCAPTCHA project creation.

Secret Key

Key generated by Google after reCAPTCHA project creation.

Theme

Defines the reCAPTCHA color. There are two options:

- **Light** :
- **Dark** :

Type

The type of reCAPTCHA. There are two options:

- **Audio**:
- **Image**:

Size

The size of reCAPTCHA. There are two options:

- **Normal** :
- **Compact**:

Position

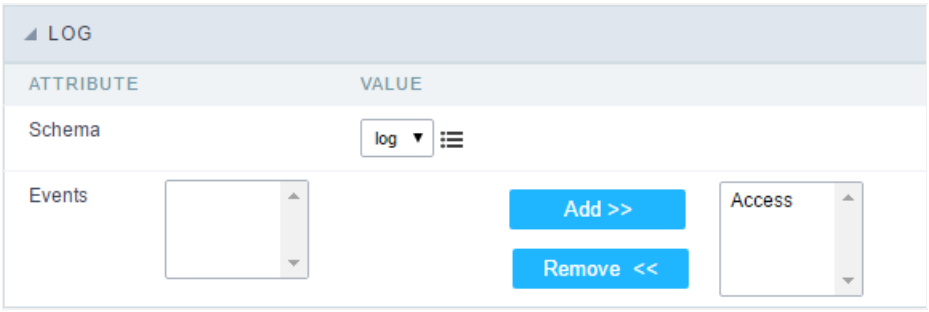
Here we can define the reCAPTCHA component alignment:

- **Left**: Position the reCAPTCHA component at the left.
- **Center**: Position the reCAPTCHA component at the center.
- **Right** : Position the reCAPTCHA component at the right.

Search Application Log

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Search - Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid	✓	✓
Procedure	✓	✓
Chart	✓	✓
Form	✓	✓
Control	✓	✓
Calendar	✓	✓
Search	✓	✓
PDF Report	✓	✓

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven't manage fields.

Related Links

Related Video ▷

- [earch with relative periods](#)
- [Dynamic Search](#)
- [Advanced search with a predefined initial value](#)
- [Grid's Quick, Refined, Advanced and Dynamic search](#)
- [Quick search](#)
- [Refined Search](#)

Search - Text Field

General Settings

This type of field allows the developer to create quickly inputs to insert and update data, where the final user can inform its data to be allocated in its database.

Type of Data

Defines the type of the field in the application. In this case we should select **Text**.

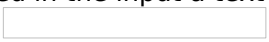
Search Label

Defines the title that will be displayed in the field when the application is executed. The terminology used in the interface is fundamental to the system have a good usability, whe should use names and familiar terms to the final user of the application, instead of using terms from the system.

For example, this text field that has the name **sellername_, the client would have a much better understanding of the functionality of the field when we define the label as __Seller name**.

Not only a fixed text, the **Label** attribute allows the use of langs to define the field's title, making it possible to use your application in a multi language project.

Watermark

Informing a text to the **Watermark** it will be displayed in the input a text as an example that can be informed in the field. The result after set will be this: 

Amount of Characters

Allows to set the width of the text field's input that varies with the amount of characters informed. Although, if the amount of characters typed are greater than the setting, the text will be pushed to the left, to keep the maximum amount of characters as defined.

Maximum Size:

Allows the user to determinate the maximum size used in the application field.

SQL Type

Informs the type of the field in the database.

Field Behavior

Use autocomplete:

Field automatically turns into autocomplete according to existing values in the database.

Use Select2

Uses the new component for data selection, allowing searches within the select comboBox.

Width for the Select2

Sets a width for the area in the Select2.

Amount of characters

Sets the amount of characters to start the search.

Amount of rows

Sets the maximum number of rows to list the search result.

Width:

Defines a width in pixels for a result box.

Search options:

Defines the validation that will be made to fetch the search result.

Start equals to: Will return the records with the same start value as in the database.
Any part: Will return the records when exist the character in any part of the record.
End equals to: Will return the records with the same final value as in the database.

Position between values:

Defines the position that objects will be displayed.

Text Between Values

Text that will appear when using a filter condition between two values.

OnChange Submit:

Submit search on this field changing.

Show Condition:

To show or not the search condition. It only works if the search has at least one condition.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).

- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Integer Field

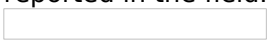
On this page, you can configure settings related to your field of type Number. Since the use of specific symbols display until the mode in which they are displayed. And thus, boost your application.

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Watermark

Informing a text for the **watermark** it will be displayed on the input text with an example of what can be reported in the field. The result after that will be configured:



Use slider:

It displays a slider component in the field. So you can increase or decrease the value sliding the cursor. You can also customize the increment value, if it increments the value 1 by 1, 2, 5, 10... N.

Amount of characters

It allows you to set the width of the input text field according to the amount of characters. However, if the quantity entered is greater than the set for the characters, the text will be pushed to the left, in order to ensure the maximum amount of characters set in the option of **Values formatting**.

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Use autocomplete:

The field turns into autocomplete automatically according to the values in the database.

Amount of characters:

Sets the amount of characters to start the search.

Number of rows:

It sets the maximum number of rows to list the search result.

Width:

it sets the width in pixels for the results.

Search options:

It sets the validation that will be made to get the search result.

Position between values:

It sets the position that objects will be displayed.

Text between values:

Text that will appear between the values.

Submit in Onchange:

Allows you to submit the filter when there are changes in the field.

Display Condition:

Allows you to display or not the condition of the filter, it will only work if the filter has at least an option.

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:

Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

```
SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}
```

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information

to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.

- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Search - Decimal Field

General Settings

Decimal field Configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Field Behavior

Use autocomplete:

Field automatically turns into autocomplete according to existing values in the database.

Use Select2

Uses the new component for data selection, allowing searches within the select comboBox.

Width for the Select2

Sets a width for the area in the Select2.

Amount of characters

Sets the amount of characters to start the search.

Amount of rows

Sets the maximum number of rows to list the search result.

Width:

Defines a width in pixels for a result box.

Search options:

Defines the validation that will be made to fetch the search result.

Start equals to: Will return the records with the same start value as in the database.
Any part: Will return the records when exist the character in any part of the record.
End equals to: Will return the records with the same final value as in the database.

Position between values:

Defines the position that objects will be displayed.

Text Between Values

Text that will appear when using a filter condition between two values.

OnChange Submit:

Submit search on this field changing.

Show Condition:

To show or not the search condition. It only works if the search has at least one condition.

Values format

Decimal Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator, Negative sign and Negative number format.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Currency Field

General Settings

Currency field Configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Currency, you can currency values to the field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

Currency Field Behavior Interface of the Search Configuration.

- **Use auto-complete** : The field behaves as an auto-complete according to the values existing in the database.
- **Amount of characters** : Sets the amount of characters to start the search.
- **Amount of rows** : Sets the maximum number of rows to list the search result.
- **Width** : Sets the width in pixels for the result box.
- **Search options** : Defines the validation that will be made to fetch the search result.
- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Values format

Currency Field Format of Values with Regional Settings.

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).

- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Date Field

General Settings

- **Data Type** : Define the type of field for the application. When set to Date, you can inform a date.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Watermark**: Displays a watermark in the field input.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the datatype of field in the database.

Values format

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator attribute.
- **Date separator** : Allows you to inform the separator symbol for the date.
- **Display** : Select the format of the day for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **Date**. You need to use the characters **A**, **M** and **D** that correspond to **Year**, **Month** and **Day**.
- **Use Combo-box** : Allows you to select the date using a combo-box.
  - **Year as Combo** : Allows to use the year combo to select the date.
    - **Initial Year** : First year displayed in the combo.
    - **Actual Year +** : Display the current plus the amount of years informed.
- **Month in full textual** : Displays the Month format in Full.
- **Display Calendar** : Enables the a calendar icon beside the field, this allows to select the date from a calendar with the format already setup.
  - **New Calendar** : Defines if the JQuery calendar (New Calendar) is going to be displayed or the old format.
  - **Years Limit** : Amount of years displayed in the calendar.
  - **View week number** : Displays the number of the week in the application.
  - **Additional months** : Displays the additional months of the calendar.
  - **Show Combo year and month** : Displays the year and month of the calendar in the combo box.

Field Behavior

- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Special Conditions

- **All Period** : Searches for all periods of dates.
- **Today** : Searches in today’s date.
- **Yesterday** : Searches in yesterday’s date.
- **Last 7 days** : Searches the last 7 days. Ex: ((01/01/2017 01/07/2017).
- **This month** : Searches the dates from the first day of the current month.
- **Last month** : Searches the dates from the first day of lasts month.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of

Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Search - Time Field

General Settings

![Time field Configuration Interface.][filtro_cons_hora] *Time field Configuration Interface.*

- **Data Type** : Define the type of field for the application. When set to Time, you can inform a time to this field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Values format

![Time Field Format of Values with Regional Settings.][filtro_cons_hora_filtro] *Time Field Format of Values with Regional Settings.*

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator and time separator attributes.
- **Time separator** : Allows you to inform the separator symbol for the time.
- **Display** : Select the format of the time for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **TIME**. You need to use the characters **HH**, **II**, and **SS** that correspond to **Day**, **Hour**, **Minutes** and **Seconds**.

Field Behavior

![Time Field Behavior Interface of the Search Configuration.][filtro_cons_hora_format] *Time Field Behavior Interface of the Search Configuration.*

- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.

- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Datetime Field

General Settings

![Datetime field Configuration Interface.][filtro_cons_data_hora] *Datetime field Configuration Interface.*

- **Data Type** : Define the type of field for the application. When set to Datetime, you can inform a date and time to this field.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Values format

![Datetime Field Format of Values with Regional Settings.][filtro_cons_data_hora_format] *Datetime Field Format of Values with Regional Settings.*

- **Regional Settings** : Allows to apply the Regional Settings to format the dates of the field. When not enabled, it will be displayed the date separator and time separator attributes.
- **Date separator** : Allows you to inform the separator symbol for the date.
- **Time separator** : Allows you to inform the separator symbol for the time.
- **Display** : Select the format of the day/time for display.
- **Internal Format** : Allows to define the format the field when the SQL type is different from **DATETIME**. You need to use the characters **A**, **M**, **D** , **HH**, **II**, and **SS** that correspond to **Year**, **Month**, **Day**, **Hour**, **Minutes** and **Seconds**.
- **Use Combo-box** : Allows you to select the date using a combo-box.
  - **Year as Combo** : Allows to use the year combo to select the date.
    - **Initial Year** : First year displayed in the combo.
    - **Actual Year +** : Display the current plus the amount of years informed.
- **Month in full textual** : Displays the Month format in Full.
- **Display Calendar** : Enables the a calendar icon beside the field, this allows to select the date from a calendar with the format already setup.
  - **New Calendar** : Defines if the JQuery calendar (New Calendar) is going to be displayed or the old format.
  - **Years Limit** : Amount of years displayed in the calendar.
  - **View week number** : Displays the number of the week in the application.
  - **Additional months** : Displays the additional months of the calendar.
  - **Show Combo year and month** : Displays the year and month of the calendar in the combo box.

Field Behavior

![Date Field Behavior Interface of the Search Configuration.][filtro_cons_data_hora_format] *Date Field Behavior Interface of the Search Configuration.*

- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Special Conditions

![Datetime Field Special Conditions.][filtro_cons_data_cond] *Datetime Field Special Conditions.*

- **All Period** : Searches for all periods of dates.
- **Today** : Searches in todays date.
- **Yesterday** : Searches in yesterdays date.
- **Last 7 days** : Searches the last 7 days. Ex: ((01/01/2017 01/07/2017).
- **This month** : Searches the dates from the first day of the current month.
- **Last month** : Searches the dates from the first day of lasts month.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**

- **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Select Field

General Settings

![Select field Configuration Interface.][filtro_cons_select] *Select field Configuration Interface.*

- **Data Type** : Define the type of field for the application. When set to Select, you can select multiple option from a combo box (Select Field).
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

![Select Field Behavior Interface of the Search Configuration.][filtro_cons_cpf_filtro] *Select Field Behavior Interface of the Search Configuration.*

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

  Selecting the lookup type.

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimitation.

      - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Height** : Defines the height for the select object.
    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to

a value informed manually (Title internal value).

- **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
- **Title** : Text displayed on the title of the Select object.

- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.
- **Link** : Allows to create a link to another form allowing to manipulate the list displayed on the select field. After the manipulation, the select object it updated automatically.

◦ **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Height** : Defines the height for the select object.
  - **Multiple Values (delimiter)**

    You can store various values for the select field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Height** : Defines the height for the select object.
  - **Multiple Values (position)**

    Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of

bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Height** : Defines the height for the select object.

- **Multiple Values (binary)**

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

Editing Lookup Configuration Multiple Values (Binary)

- **Label** : Text that will be displayed in the item list of the select field.
- **Height** : Defines the height for the select object.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  Displaying the original lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Lookup Method - Actual value

This lookup is used to list all the values in the selected field.

This lookup will apply a “distinct” to your SQL query.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.

- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Double Select Field

General Settings

![Double Select field Configuration Interface.][filtro_cons_duplo_select] *Double Select field Configuration Interface.*

- **Data Type** : Define the type of field for the application. When set to Double Select, your allowed to have multiple options selected.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

![Double Select Field Behavior Interface of the Search Configuration.][filtro_cons_cpf_filtro] *Double Select Field Behavior Interface of the Search Configuration.*

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, getting these values from the database.

Lookup Settings Display for the field. *Automatic Lookup Interface..*

- **SQL Select Statement** : Defines the SQL command that will get the values displayed on the form field. The command needs to have the following structure:

  ```
  SELECT key_field , display_field FROM table
  ```

  The value of the key_field will be stored in the table field.

- **Height** : Set the height(lines) of the field interface.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter than (;).
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**

- **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Checkbox Field

General Settings

![Check box field Configuration Interface.][filtro_cons_checkbox] *Check box field Configuration Interface.*

- **Data Type** : Define the type of field for the application. When set to Check box, your allowed to have multiple options selected.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

![Check box Field Behavior Interface of the Search Configuration.][filtro_cons_cpf_filtro] *Check box Field Behavior Interface of the Search Configuration.*

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query). *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the CheckBox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
    - **Option check all** : Displays two options on the field to check and uncheck all.
    - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Columns** : Set the amount of columns, for the list of items.
  - **Multiple Values (delimiter)**

    You can store various values for the checkBox field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Columns** : Set amount of columns, for the list of items.
  - **Multiple Values (position)**

    Stores a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

    For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

    The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

    Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the

following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Columns** : Set the amount of columns, for the list of items.

- **Multiple Values (binary)**

  Stores a diversity of information selected on the field in only one value.

  In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  To store the data, it is done an internal sum of the decimal number. See the examples below.

  Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 2 (Culture) = **3**

  Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

  Setting up Multiple Values (Binary)

  - **Label** : Text that will be displayed in the item list of the checkBox.
  - **Columns** : Allows you to inform the amount of columns, for the list of items.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Saves all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Refreshes the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Option check all** : Displays two options on the field to check and uncheck all.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio,

Select, Text, etc
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Radio Field

General Settings

![Radio field Configuration Interface.][filtro_cons_radio] *Radio field Configuration Interface.*

- **Data Type** : Define the type of field for the application. When set to Radio, your allowed to select one of the options listed.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

![Radio Field Behavior Interface of the Search Configuration.][filtro_cons_cpf_filtro] *Radio Field Behavior Interface of the Search Configuration.*

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query). *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Columns** : Allows you to inform the amount of columns, for the list of items.

    - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

      - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
      - **Title** : Text displayed on the title of the Select object.

- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

  This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

  Manual Lookup Interface.

  - **Label** : Text that will be displayed in the item list of the radio.
  - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
  - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
  - **Insert Button** : Adds to the list the values informed on the fields Label and Value.
  - **Update Button** : Modifies the attributes of the selected item.
  - **Remove Button** : Remove the selected item from the list.
  - **Clear Button** : Clear the Fields.
  - **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on load lookup definition.
  - **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
  - **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

    - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
    - **Title** : Text displayed on the title of the Select object.
  - **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
  - **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

    Displaying the original lookup.
  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Text Auto-Complete Field

General Settings

![Text Auto-Complete field Configuration Interface.][filtro_cons_texto_auto] *Text Auto-Complete field Configuration Interface.*

- **Data Type** : Define the type of field for the application. When set to Text auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal Text for the data.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

![Text Auto-Complete Field Behavior Interface of the Search Configuration.][filtro_cons_texto_filtro] *Text Auto-Complete Field Behavior Interface of the Search Configuration.*

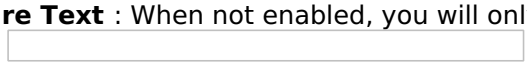
- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **Case Settings** : Allows to convert the letters of the field when losing the focus. The options are:
  - **Upper Case** : All in Upper Case.
  - **Lower Case** : All in Lower Case.
  - **Capitalize first word** : Capitalizes the first letter of the word.
  - **Capitalize all words** : Capitalizes the first letter of all the words.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Search Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below.  *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

Help type - Pop-up configuration Interface.

- **Hint** : Passing the cursor over the field, you will view a hint with the help description.

Help type - Hint configuration Interface.

- **Text** : It will display the help description beside the field.

Help type - Text configuration Interface.

Search - Number Auto-Complete Field

General Settings

![Number Auto-Complete Field Behavior Interface of the Search Configuration.][filtro_cons_número_auto]
Number Auto-Complete Field Behavior Interface of the Search Configuration.

- **Data Type** : Define the type of field for the application. When set to Number auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal number for the data.
- **Search Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Use same label used on the Grid** : When this option is enabled, it allows to field to use the same title defined in the grid, in this case the Search Label is not used.
- **Amount of characters** : Define the amount of characters allowed for the field.
- **Maximum Size** : Defines the size of the field.
- **SQL Type** : Informs the data type of field in the database.

Field Behavior

![Number Auto-Complete Field Behavior Interface of the Search Configuration.][filtro_cons_decimal_filtro]
Number Auto-Complete Field Behavior Interface of the Search Configuration.

- **Use autocomplete** : The field behaves as an autocomplete according to the values existing in the database.
- **Position between values** : This option sets the position that objects will be displayed.
- **Text between values** : Text that will appear between values.
- **OnChange Submit** : When enabled, it submits the search when there is modifications to the field.
- **Show Condition** : When enabled, it displays the condition of the search in the Grid, it will only work if there is at least one option selected.

Values format

![Number Auto-Complete Field Format of Values with Regional Settings.][filtro_cons_decimal_format] *Number Auto-Complete Field Format of Values with Regional Settings.*

- **Regional Settings** : Allows to apply the Regional Settings to format the number of the field. When not enabled, the attributes Digit Grouping Separator, Decimal Separator, Negative sign and Negative number format.
- **Decimal Precision** : Defines the amount of decimal places for the field.
- **Digit grouping** : Defines if the field will display the digits separator.
- **Decimal Separator** : Defines the decimal separator.
- **Accept** : Determines if the field, will accept only negative, positive or both numbers.

Search Lookup

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).

- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below. *lookup Settings configuration Interface.*

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

lookup Settings validation configuration Interface.

- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

This feature allows that the instructions are documented to be used for the generated application, this helps

the end user to understand better how the system works.

- **Help Description** : Allows to inform a text that will present to the user when he positions the mouse over the field.
- **Help Type**
  - **Pop-Up** : Selecting the pop-up type, it will display an icon beside the field that when clicked, you will view a pop-up with help description.

    Help type - Pop-up configuration Interface.

  - **Hint** : Passing the cursor over the field, you will view a hint with the help description.

    Help type - Hint configuration Interface.

  - **Text** : It will display the help description beside the field.

    Help type - Text configuration Interface.

Search Application Events Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application's CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event's PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Search

This event occurs just once before it display the search form. It can be used to set some default values to the fields. For example: {company} = [glo_company]; Date fields are used differently, They use a suffix "day" for day, "month" for month and/or "year" for year. For example, The {Birth} field will be: {Birth_day}=date("d"); {Birth_month}=date("m"); {Birth_year}=date("Y");

onScriptInit - Search

This event occurs when the application its loaded, or reload. Occurs before the application runs the SQL statement, so using this event it is possible to modify the grid SQL statement dynamically, based on any logical. Macros frequently called on this event: sc_select_field, sc_select_order, sc_select_where(add), etc.

onRefresh - Search

This event occurs when the form application is reloaded, is possible to reload the form based on fields (select , radio, checkbox)

onSave - Search

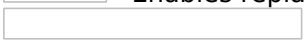
This event occurs always that some search setting is saved in the filter application.

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

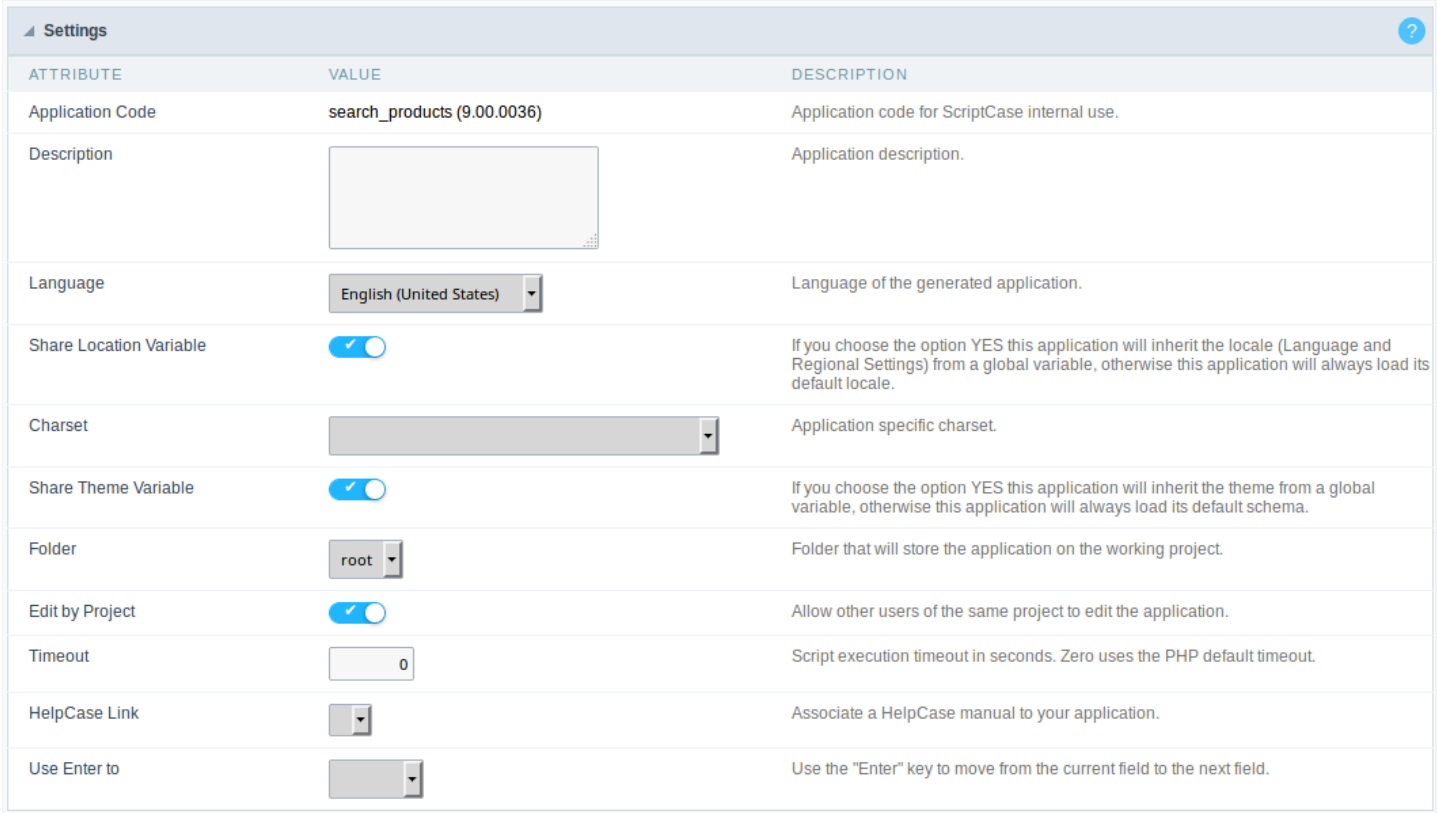
The hotkeys are available in all browsers compatible with Scriptcase.

> Click Here to view the Scriptcase hotkeys documentation.

Application Settings (Search Application)

Settings

With this interface, you can set the common attributes of the app.

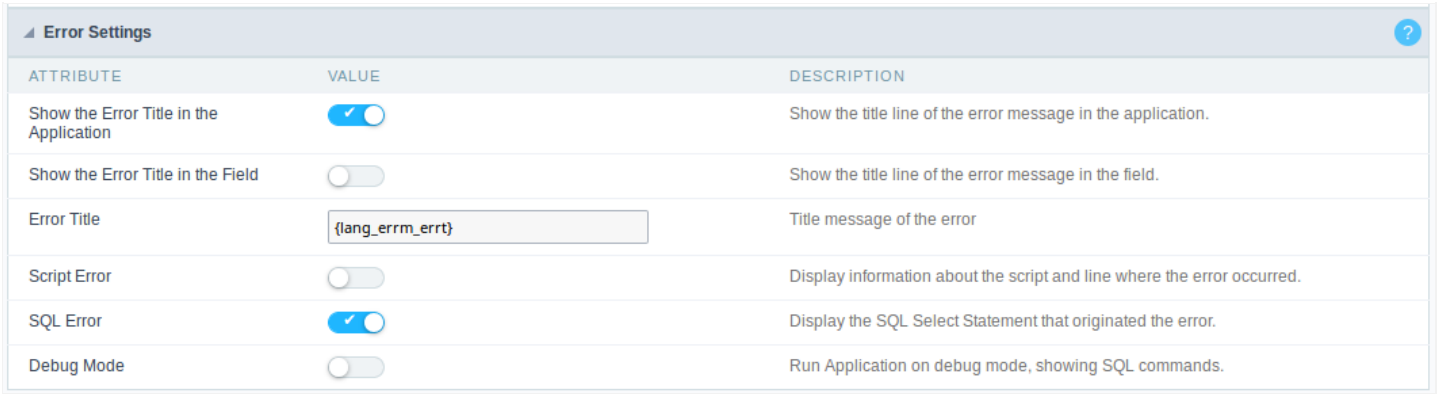


Application Settings Interface

- **Attributes**
  - **Application Code** : It is the name that defines an application. An app can be renamed at the List of Application.
  - **Description** : This field contains a brief description of the application objectives.
  - **Language** : Set the default language of the application. Display all the application hints and messages in the selected language.
  - **Share Location Variable** : Define if the app shares the regional settings with other applications through a session variable.
  - **Charset** : Define a specific charset to use in the application.
  - **Share Theme Variable** : Define if the app shares the Theme settings with other applications through a session variable.
  - **Folder** : Define the project folder that contains the app.
  - **Edit by Project** : Define if other project developers can edit the application.
  - **Timeout** : Set the session runtime timeout in seconds. If the value is Zero, it assumes the default timeout of the PHP.
  - **HelpCase Link** : It allows to associate a HelpCase file with the application.
  - **Use Enter to** : Allows selecting the action of the Enter key (submit or tabulate the Form).

Error Settings

It groups the notification options of the app.

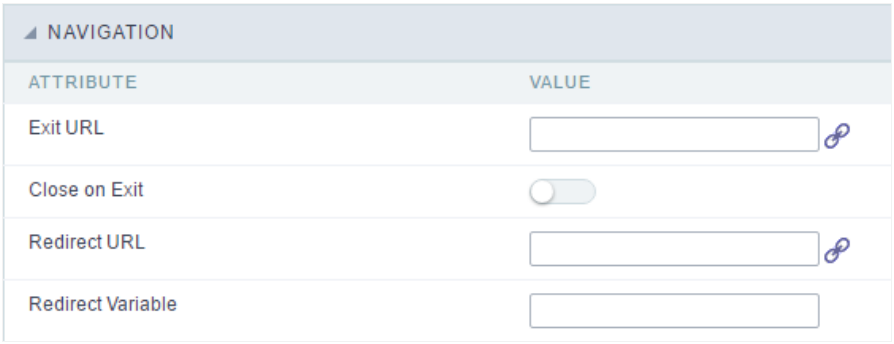


Error Settings Interface.

- **Attributes**
  - **Use SweetAlert**: Use the SweetAlert to display messages from the application. When this option is active, it will replace the browser's "confirm" and "alert".
  - **SweetAlert position using Toast** : The position to display error messages on the application.
  - **Show the Error Title in the Application** : Define to display the title line of the error message or not.
  - **Show the Error Title in the Field** : Define to display the title line of the error message in the field or not.
  - **Script Error** : Allows displaying the line code where there is an error..
  - **SQL Error** : Allows displaying the SQL statement if it got an error.
  - **Debug Mode** : Runs the application in Debug mode, showing all SQL statements the application is executing.

Navigation

This interface allows defining the navigating behavior of the application



Navigation Interface.

Exit URL

URL to where the user goes when he clicks on the "exit" button.

Close on Exit

Close the browser window when the user clicks on the "exit" button.

Redirect URL

Redirect to another URL in case there aren't any global variables available.

Redirect Variable

Creates a variable with the application name and sends it to the redirected application.

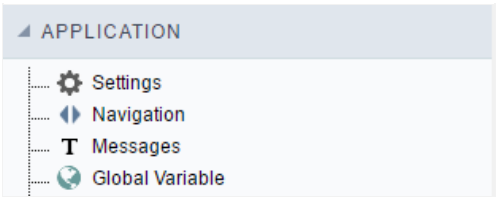
Global Variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target="_blank"} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



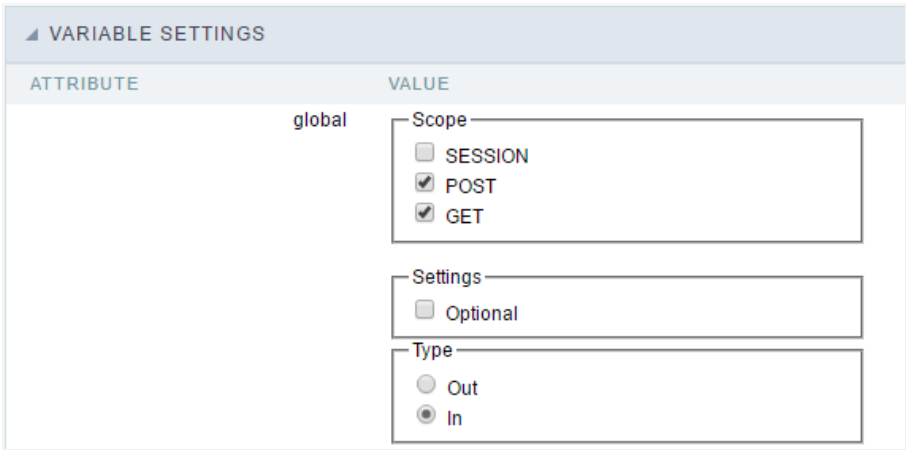
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

Indicates where the application is using the variable.

Search Application Links Overview

This feature allows the developer to create links between applications of the same project, expanding the integration of applications. All link options are grouped under the Application Links menu.

Application Links

In the first access to the menu, we can see the list of existing connections in the application.

> If the application does not have a configured link, the application list screen will be displayed with the message: **This application does not have any link. Click here to create one now.**

ID

Link identification ID.

Type

The type of link created, some links such as Edit Link allow only one link per application. In this case, the developer will be able to check the connection types that already exist in the application

Target Application

This info show the target application name.

Actions

This column has edit options for the links.

Properties

It allows accessing the binding properties where it is possible to configure the binding behavior.

Link

Displays the links screen, where it is possible to configure the link that was made with the application informed in the target application column. In this option it is possible to change the parameter passed in the connection as well as the target application.

Delete

Permanently deletes the connection in the applications.

> Through the option Restore Applications, it is possible to get a previous version of the application, making it possible to recover the link deleted.

Links Type

The Calendar application has the following links options

- Application Link - Allows you to link a control with any project application.
- Capture Link: It allows the creation of links from the filter fields of the Grid application, in order to enable the recovery of the value to fill in the field, with another Grid application of the project.

Calendar Applications Link Types x Target Application available to link

Check the links types of the Calendar application and the target applications available for each link types.

> Only **Capture link** have some restrictions in the target application use, allowing only to use of Grid

application.

	Application Link	Capture Link
Grid		
Chart		
Form		
Control		
Search		
Menu		
Tree Menu		
Tabs		
PDF Report		
Dashboard		
Blank		
Calendar		

Application Link

Creating an Application Link

This type of link allows the developer to create a link from a grid to a form with the objective of editing the register of a grid's row.

In the link options, we will choose the **Application Link**. When we choose this option, edit a register from a Grid will be possible.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Open in an iframe:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a iframe in one of the four options available in the iframe settings.

Open in a parent:

When we use this option the target application will be displayed in the same window of our application, and the target application will have a back button so we can return to the previous application.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Display the button new in the grid:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Also, when we select the option **Yes**, it is displayed two new options **Label to the New button** and **Hint to the New button**:

Label to the New button:

In this option we will inform a name for the button that will be displayed in the Grid.

Hint to the New button:

In this option we will inform a message that will be displayed when the mouse cursor is over the **New** button.

Shortcut key to the New button:

Indicates the shortcut key to the button add new register.

Display the button New in the grid:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Shortcut key to the New button:

Indicates the shortcut key to the button add New Register.

Label to the New button:

In this option we will inform a name for the button that will be displayed in the Grid.

Hint to the New button:

In this option we will inform a message that will be displayed when the mouse cursor is over the **New** button.

Open in an iframe.

When we use this option the target application will be displayed in the same window of our application, and

the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Shortcut key to the New button:

Indicates the shortcut key to the button add New Register.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Iframe properties

In this settings, we can define some iframe details that will display the target application.

Display the target application's header:

When the option **Yes** is selected, the header of the target application is also displayed in the iframe.

Iframe position in relation to the main application:

In this option, we will define in which position in relation to the main application the iframe will be displayed, there are four options:

Below: The iframe will be displayed below the main application.
Above: The iframe will be displayed above the main application.
Right: The iframe will be displayed on the right of the main application.
Left: The iframe will be displayed on the left the main application.

Action after an insert:

In this option, we will define what will happen after a register insert, there are two options:

Reload the grid: The current page will be refreshed after the insert.
Move to the end of the grid : After the insert will be displayed the last page of the grid will the last register inserted.

Iframe height:

Allows to set the iframe height.

Iframe width:

Allows to set the iframe width.

Display the button New in the grid:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Shortcut key to the New button:

Indicates the shortcut key to the button add New Register.

Label to the New button:

In this option we will inform a name for the button that will be displayed in the Grid.

Hint to the New button:

In this option we will inform a message that will be displayed when the mouse cursor is over the **New** button.

Form Properties

In those settings, we can define which buttons will be available in the target form application. Initially we have five options, that are:

Enable Insert button:

In this option we can define if the **New** button will be available in the target application.

Enable Update button:

In this option we can define if the **Save** button will be available in the target application.

Enable Delete button:

In this option we can define if the **Delete** button will be available in the target application.

Enable Navigation buttons:

In this option we can define if the buttons **first, previous, next and last** will be available in the target application.

Enable register editing button in the grid:

In this option we can define if the edit register button, which is the **pencil** in the grid, will be available.

Capture Link

Creating a Capture Link

The capture link is used to return a value from a **Grid** to a **Form** field.

In the types of links options, we will choose the **Capture Link**. After selection this option, we should also choose which field we want to return the value of.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Application with no parameters

However, when the target application does not have any defined parameters, the following screen is displayed:

Clicking in the button, you will be taken to the target application to create a parameter, so you can use the update button in the **parameters definitions** to refresh them.

Link properties

In this screen we will set the application display mode that will be called in the link.

In this type of link there is only one display option:

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Modal:

In this option we are going to define if the target application will be opened in a Modal.

Yes: This option will make the target application be opened in a modal. **No**: This option will make the target application be opened in a new window.

If **Yes** is selected in the previous option, the Modal **Height** and **Width** will be available.

![Modal with yes][modal2]

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

If **No** is selected in the previous option, only those options will be available.

Allows to modify manually in the update:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Allows to modify manually in the insert:

Indicates which shortcut key to the button add new register.

Allows to modify automatically in the update:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Allows to modify automatically in the insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Search Application Button link Settings

Creating a Button Link

Allows the developer to create a link where the call to the other application will be done through a button.

In the type of links options, we will choose the **Button Link**. Choosing this option it will be possible to create a link to any other application.

List of applications

After selecting this option, The list of applications to what you want to create a link will be displayed.

This screen can be viewed in the following ways:

All:

In which is possible to see all the project’s applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

![Same window][mesma_janela]

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

![Another window][outra_janela]

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open the application in a tab inside the menu:

Choosing the option **Yes**, the application will be open in a **menu tab**, when executed from a menu application, instead of opened in another browser window.

Choosing the option **Yes**, there are also some other settings to be displayed:

![Other window yes][outra_janela1]

Title to the tab:

This option allows the developer to set a title to the tab that will be opened when used in a Menu application.

Hint to the tab:

This option allows the developer to set a message to be displayed when the mouse cursor is over the tab Menu.

Active tab icon:

This option allows the developer to set an icon to be displayed in the tab when used in a menu application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

![Modal][modal]

Height:

Allows to set the Modal’s height.

Width:

Allows to set the Modal’s width.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

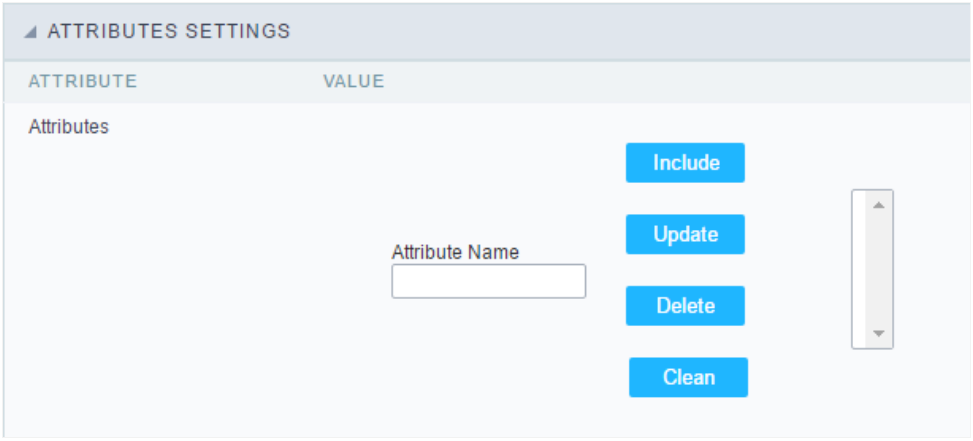
In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Search Application Programming

ScriptCase has incorporated the concept of Object Oriented programming, using attributes, resources, methods and libraries. It is possible to create your own business rules in applications, and by using these concepts you can reap huge rewards in terms of better organization and improved development.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

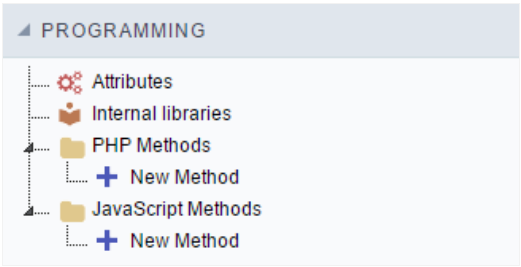
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

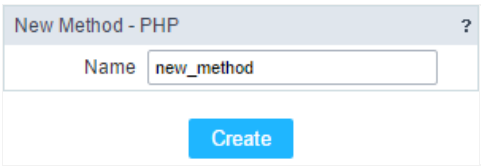
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

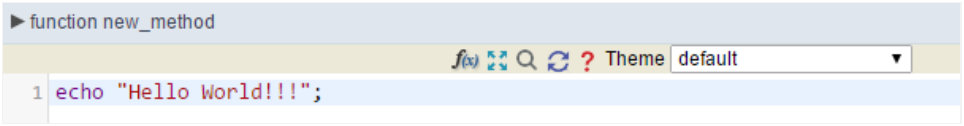


- Give the method a name and click Create. As in the image below.

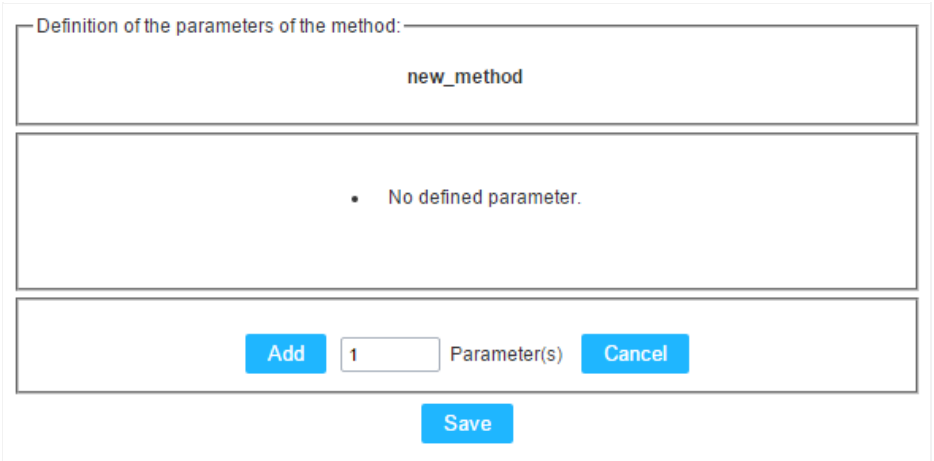
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



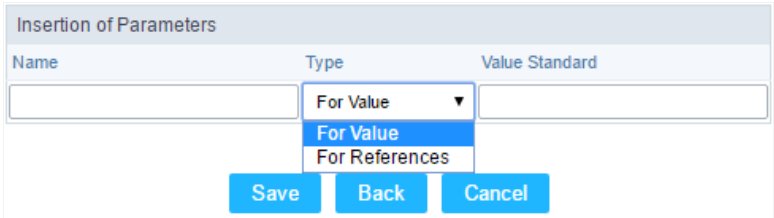
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

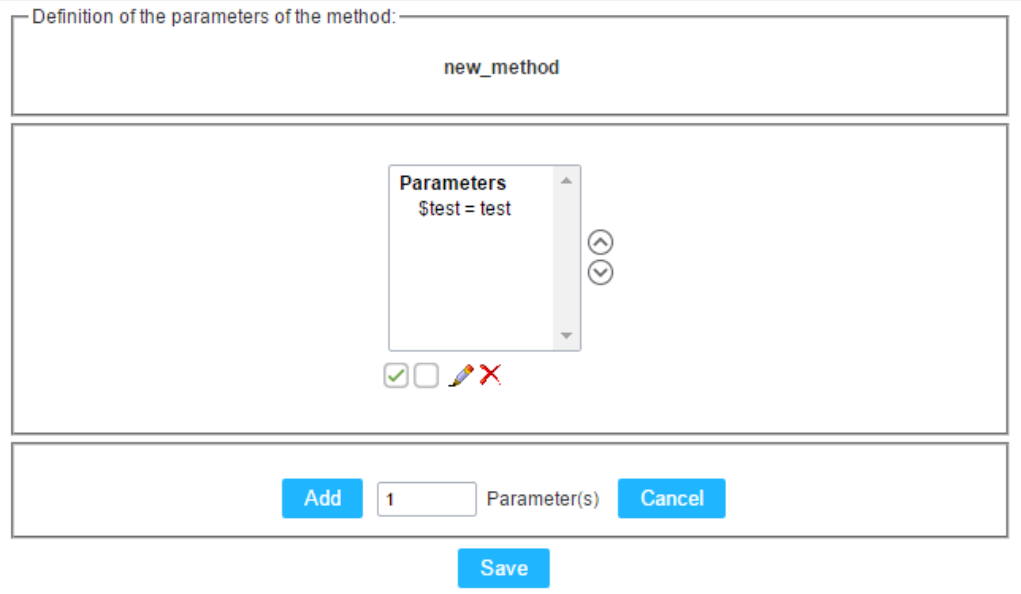


- **Name** : Type in the variable's name.
- **Type** : Selecting the type of variables: For Value or For Reference.

- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:



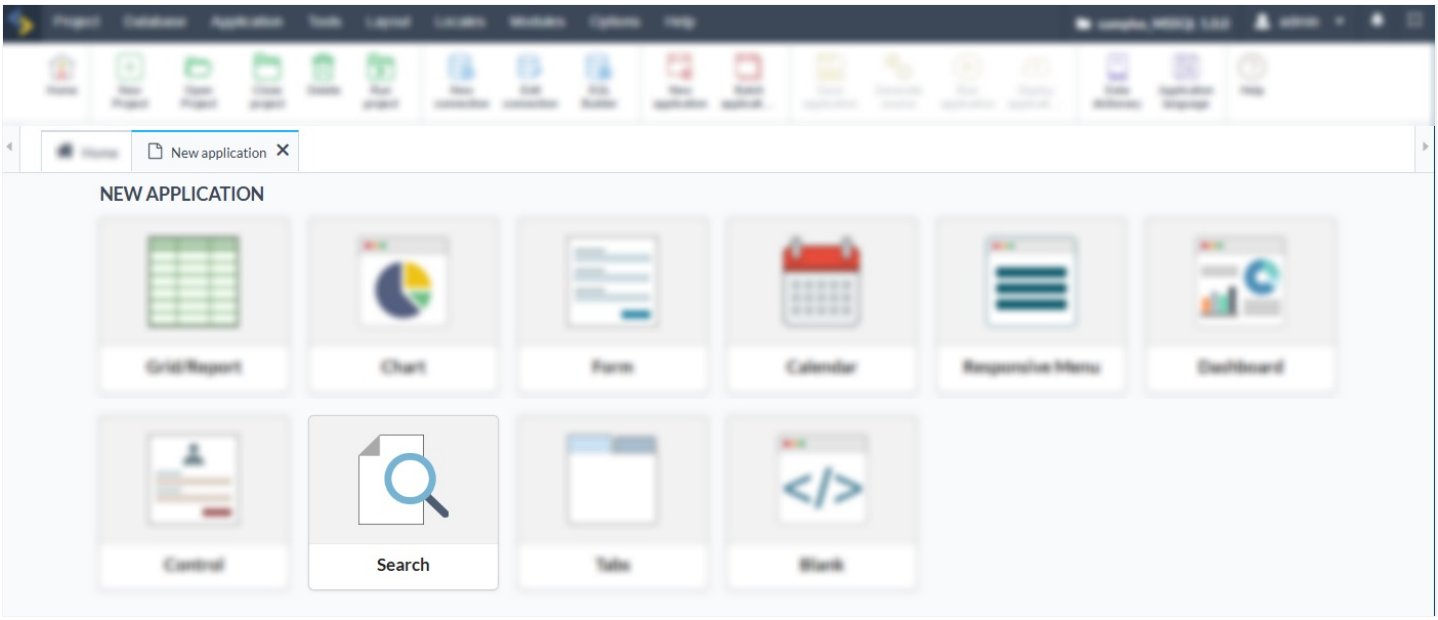
- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

Creating a Search Application

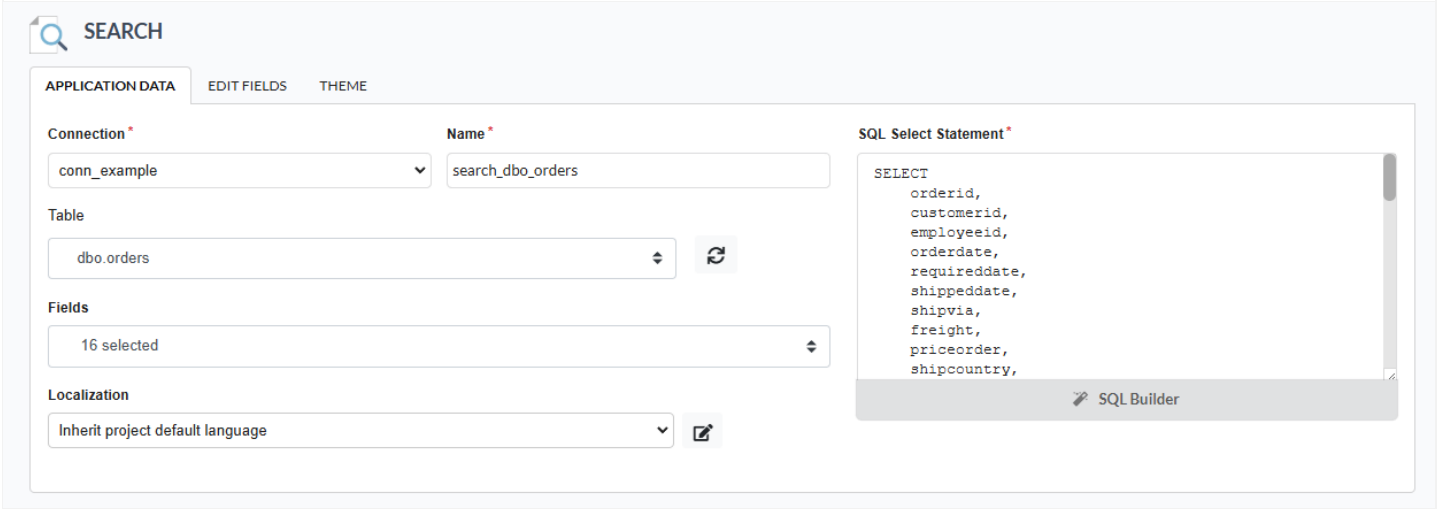
New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.



Application Data



Connection

Defines the connection used for the creation of the application. The existing connections in the project will be listed.

Name

The name of the application being created. It cannot contain special characters.

Table

Defines the tables to be used in the application. (Form and Calendar can only use one table).

Fields

Defines the fields that will be part of the applications.

Localization

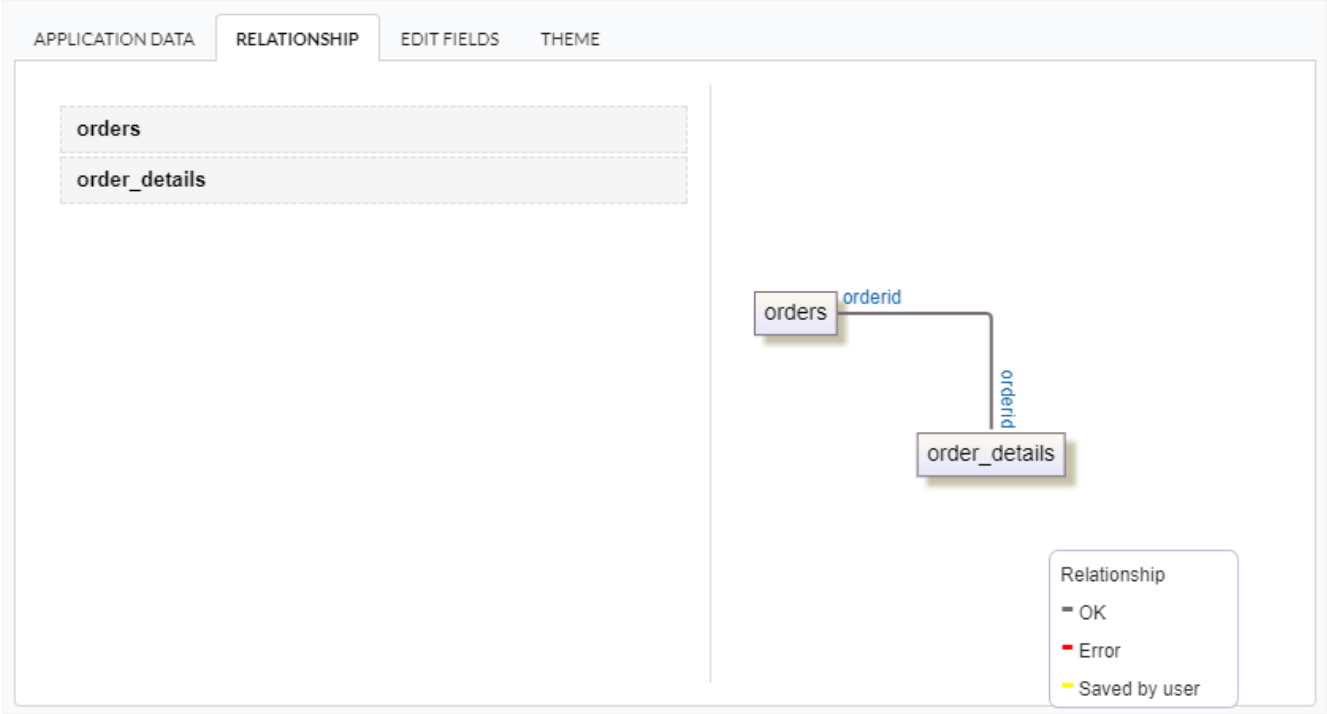
The language of the application to be created. The project's default language is automatically selected.

SQL

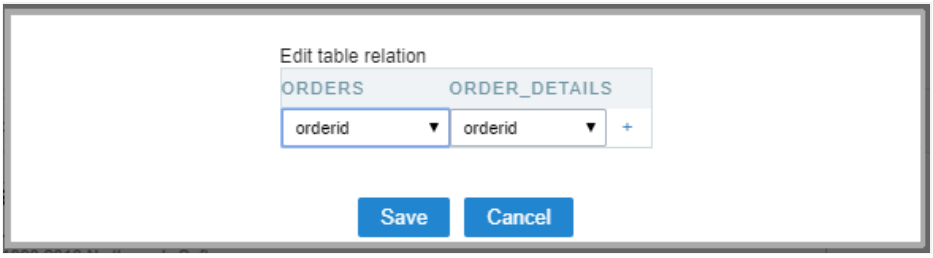
Displays the SELECT command generated after selecting the tables and fields. This field also allows the insertion of a previously created SQL statement, as long as it uses tables that exist in the database selected in the connection.

Relationship

When select two or more tables, the tab "Relationship" will be displayed. In this tab we can see the relationship created between the tables, where we can edit the related fields.

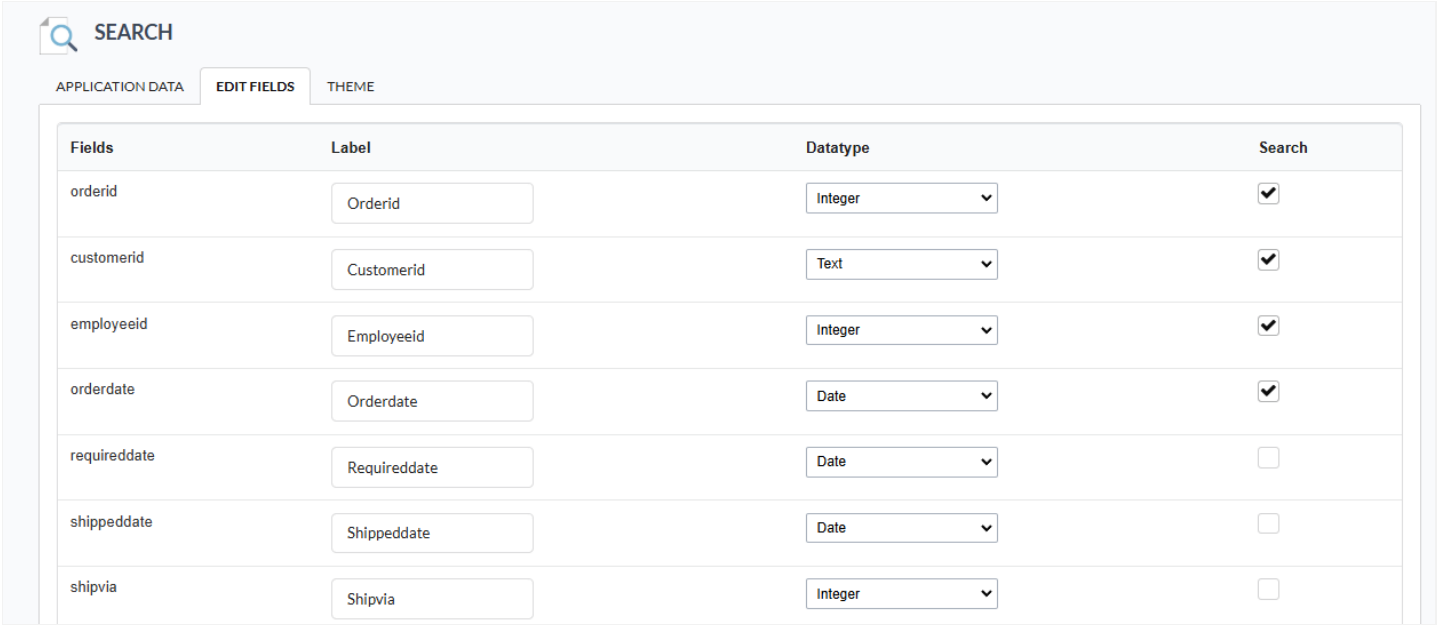


When we click in a link, in the screen above, it will displayed the related field's edition form, as you can see in the image below.



Edit Fields

This screen displays the fields of the selected tables and allows adjustments to be made before creating the application, such as changing the data type, display name, and other configurations.



Fields

Names of the database fields.

Label

Names of the fields in the generated application's interface.

Datatype

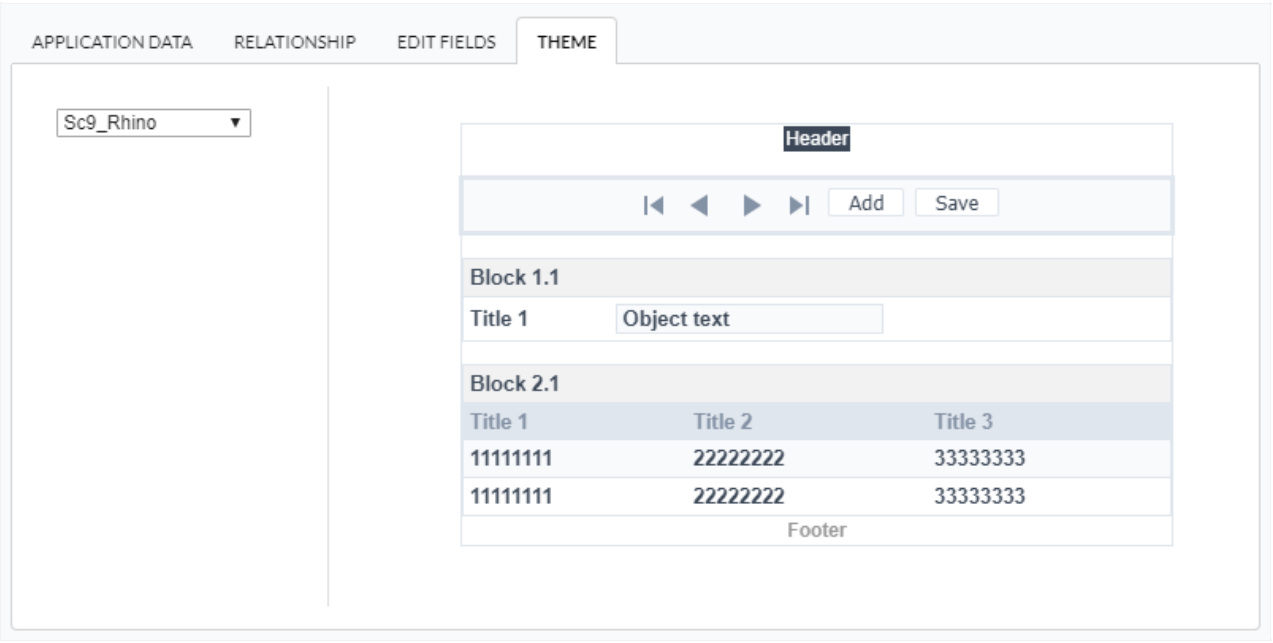
Specifies the field's data type.

Search

Defines the fields available in the filter.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.



Related Links 🔗

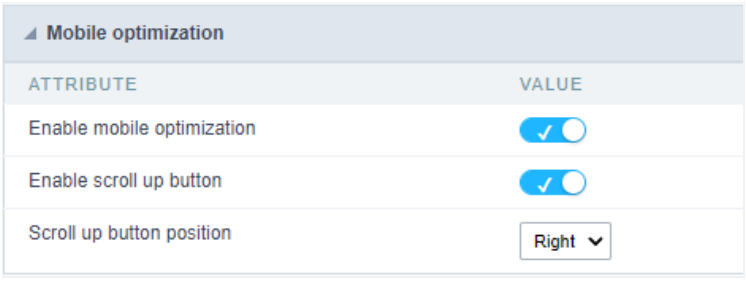
Related Video ▷

Search Application Mobile Optimization

The Mobile settings allows to define the automatic optimization of generated applications to run on mobile devices.

See below the available settings.

Enable mobile optimization



This option changes how the application HTML elements works, adapting them automatically to run on mobile devices.

It enables this option by default when creating a new application.

See some examples of adapted screen:

Enable back to top button

It enables the displaying of the button **back to top button** in the Grid, Filter, Detail and Summary modules.

It enables this option by default when creating a new application.

Back to top button position

Defines the position of the **back to the top** button:

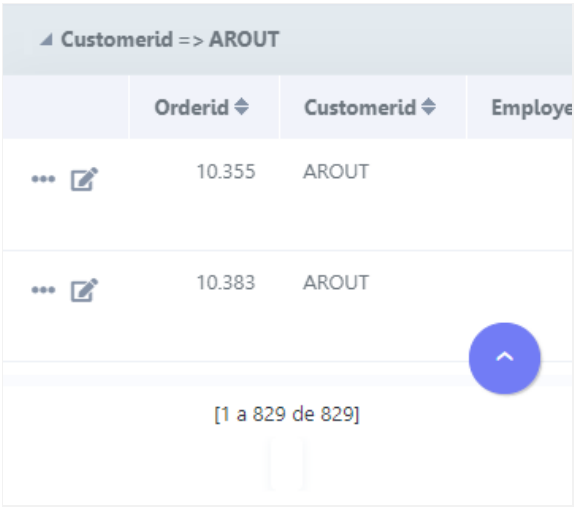
The options are:

- **Right** - Position the button in the bottom right corner;
- **Left** - Position the button in the bottom left corner;

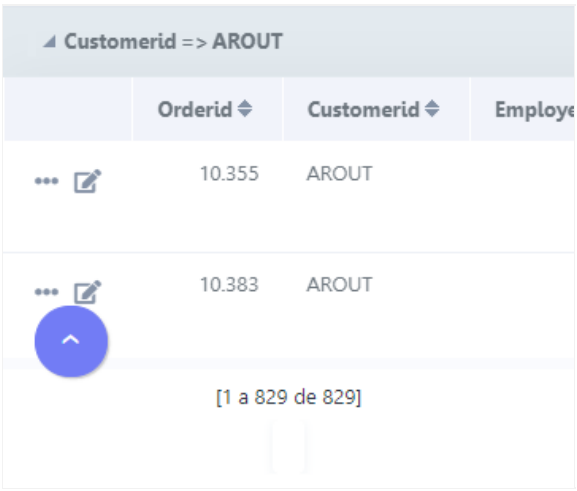
When creating a new application, the button is configured on the right by default.

See below for placement examples.

Button in the bottom right corner



Button in the bottom left corner

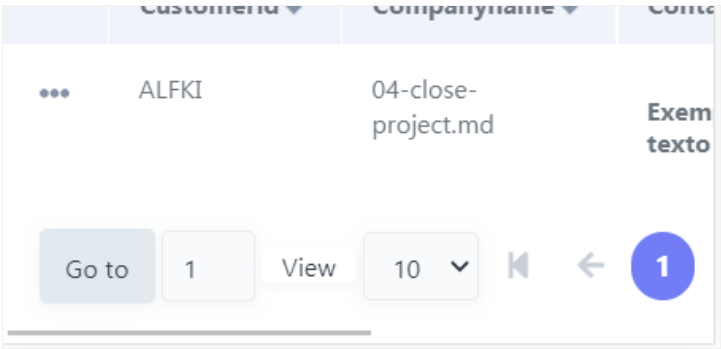


Pin the bottom toolbar

Defines the behavior of the lower toolbar when running the application on mobile devices with the **Simplified toolbar** option disabled.

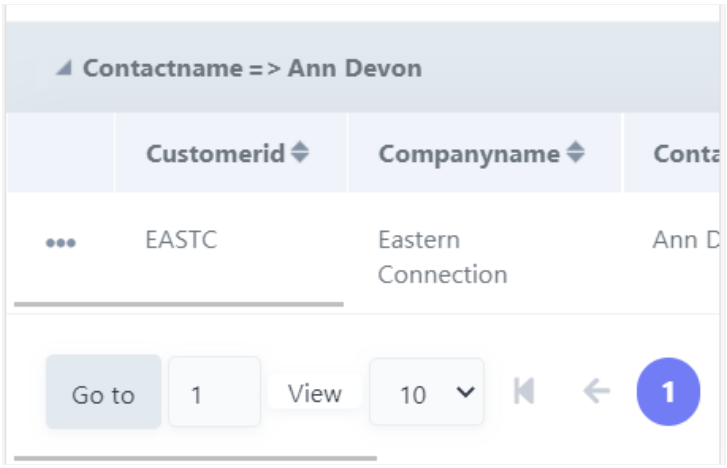
When enabling these options, the lower toolbar will behave similarly to the upper toolbar, remaining fixed on the screen and thus facilitating access to the configured resources.

Example of fixed bottom toolbar



If you choose to uncheck this feature, the lower toolbar will be fixed at the end of the application, so access to the configured buttons will only be available when scrolling to the end of the page.

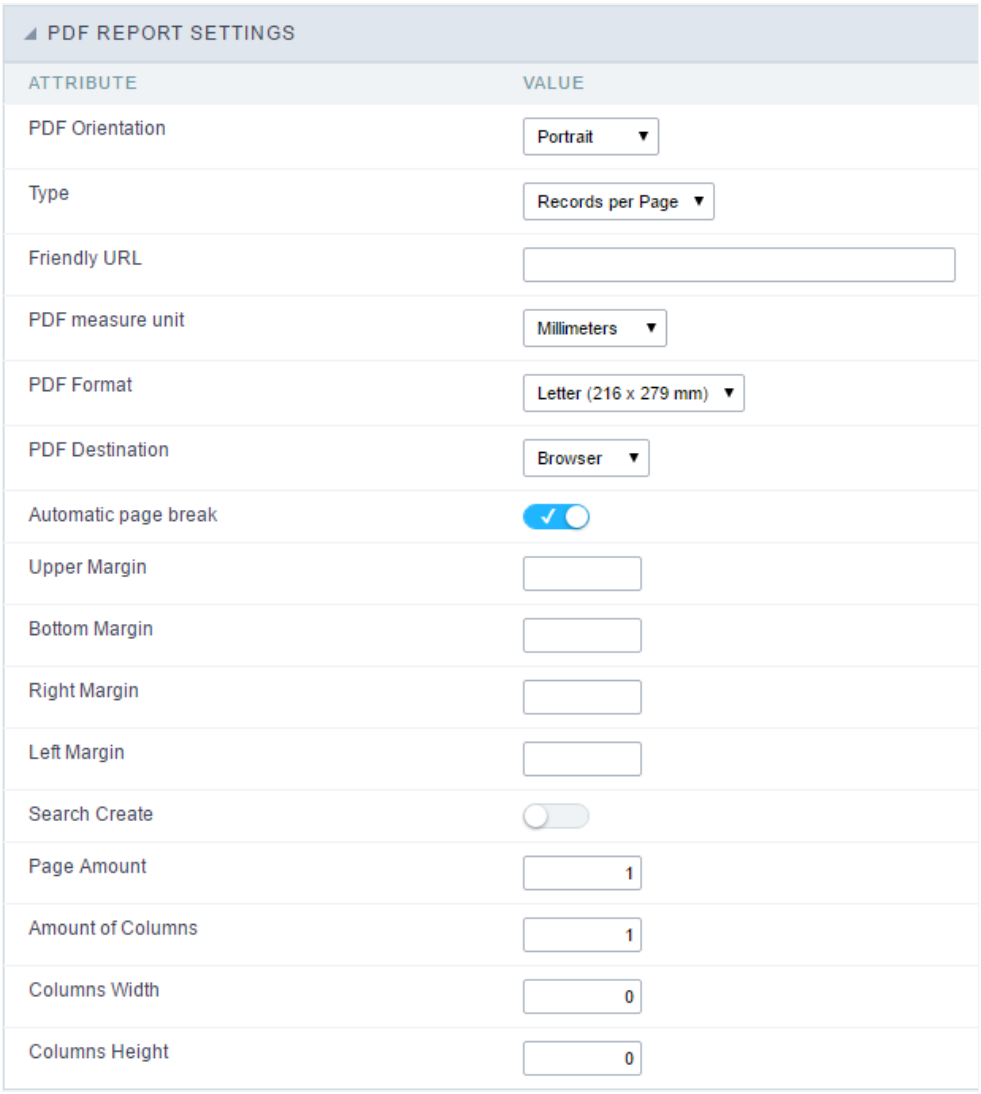
Example of Traditional Toolbar



This setting is available when the **Simplified Toolbar** options is turned off.

PDF Report Settings

PDF Report Settings



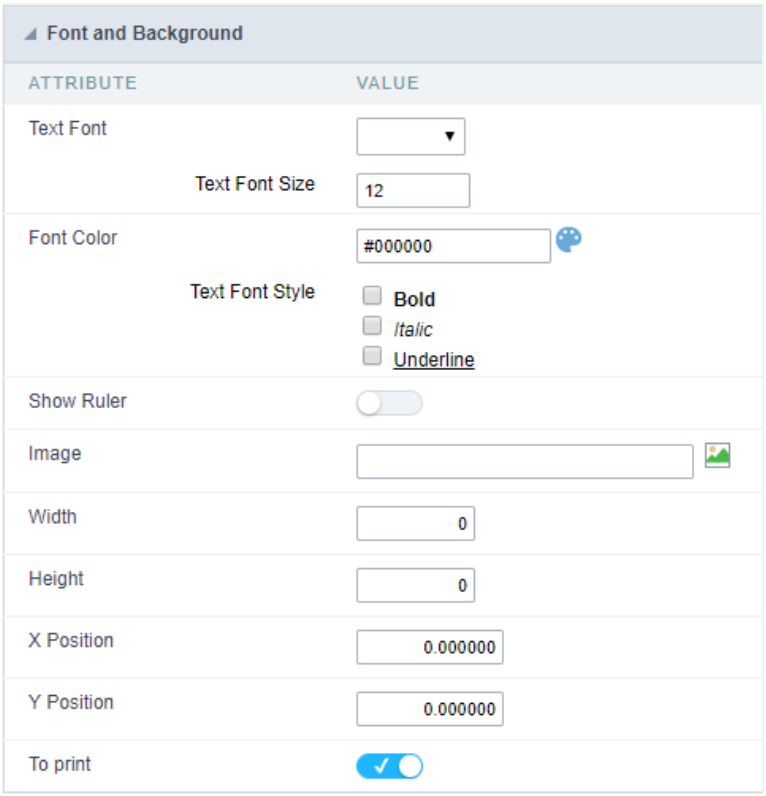
PDF Report Settings Interface.

- **Attributes**
  - **PDF Orientation** : Using this option you can set whether to print the PDF in portrait or landscape orientation.
  - **Type** : PDF type
    - **Records per Page** : Only applicable to Grids aligned horizontally, which allows you to set the number of records that will be printed per page.
    - **Records per Line** : Allows you to set the records that will be printed per page in the pdf.
  - **Friendly URL** : This field allows you to change the URL that will be called by the application. Allowed characters are the same available on URLs: a-z, A-Z, 0-9, -_. This option can also be changed on the home screen, on the “Friendly URL” column at the applications list.
  - **PDF measure unit** : Using this option you can define the unit of measurement of the position of objects in the PDF. Used in printing format, it can be: points, millimeters, centimeters and counts.
  - **PDF Format** : Using this option you can select the Form type in which the PDF will be printed (letter, A4, etc).
  - **PDF Destination** : Using this option you can set where the PDF will be sent to the Browser, Download or server.
    - **Browser** : If you use this option the PDF will open within the browser.
    - **Download** : The PDF will open for download. you must inform the name of the file to be generated. E.g.: file.pdf
    - **Server** : The PDF will be sent to the server. It is necessary to inform the path of the file to be generated. E.g.: c:\folder\file.pdf
  - **Automatic page break**: Use automatic page break on pages.

- **Margins** : Allows you to view the application on the page according to the values reported in the margins (right, left, top and bottom) in millimeters.
- **Search Creation** : Using this option you can set the PDF startup mode by the Search. So the initial mode will be a Search application.
- **Page Amount** : Allows you to set the number of pages that will be printed in the PDF.
- **Amount of Columns** : Allows you to set the number of columns that are printed, what allows you to create applications for issuing of labels printing for example.
- **Columns Width** : Allows you to set the width of the columns in the application.
- **Columns Height** : Allows you to set the height of the columns in the application.

Font and Background

Configuring the PDF Preview allows you to set values of text and layout preview.



PDF Report visualization interface settings.

- **Text Font** : Here you can choose the font family to be used initially by the Report PDF.
- **Text Font Size** : Here you can choose the font size .
- **Font Color** : Here you can choose the font color .
- **Text Font Style** : Allows you to choose the style of the text, if this will be bold, italic, or underlined, or any other combination.
- **Show Ruler** : Sets whether the PDF printing displays ruler at the edges of the pages.
- **Image** : Allows you to insert a background image on the page. This image will be also printed, it can be a watermark for your document for example.
- **Width** :Sets the image width, in pixels, to be inserted into the page.
- **Height** :Sets the image height, in pixels, to be inserted into the page.
- **X Position** : Sets the abscissa of the initial image position.
- **Y Position** : Sets the ordinate of the initial image position.
- **Print** : This option sets whether the image will be printed on the application.

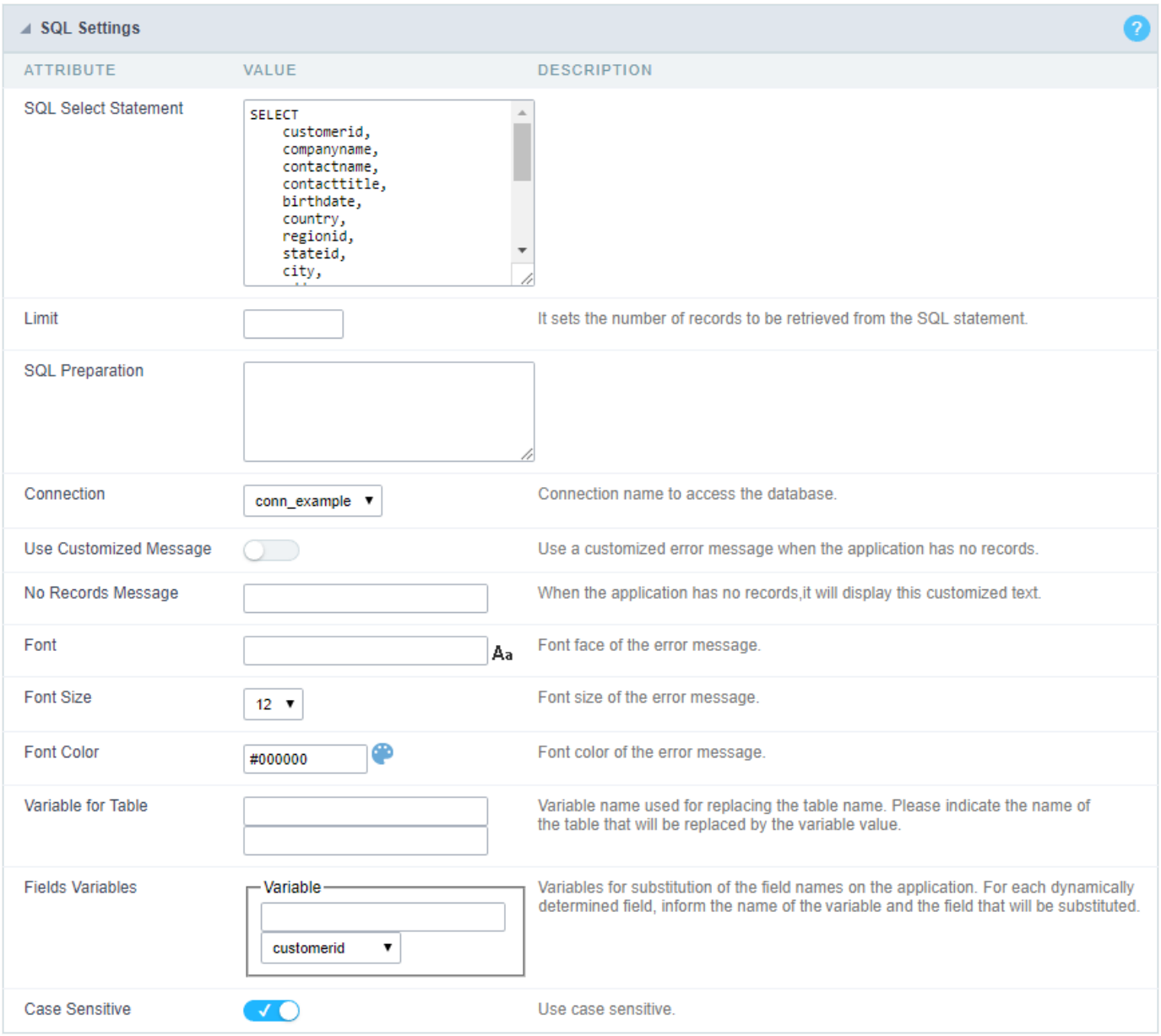
Related Links 🔗

Related Videos ▷

PDF Report SQL Settings

SQL Settings

This interface allows configuring the related database settings, such as the SQL statement, the used database connection, case sensitive, and others.



SQL configuration

SQL Select Statement

It allows you to define the primary SQL of the application. You can edit this SQL to add or delete fields.

Limit

Lets you limit the display in the number of records retrieved by SQL query.

SQL Preparation

You can enter SQL commands or procedure names to execute them before the primary SQL of the application.

Grid

Connection

It allows defining the database connection of the application. You can change the connection to another one that has the same table.

Use Customized Message

Lets you define to display the “no records” message or not.

No Records Message

Lets you set the message when the application has no records.

Font

This option is available when using the “Use Customized Message” option. It lets you set the font for the message.

Font Size

This option is available when using the “Use Customized Message” option. It lets you to set the font size.

Font Color

This option is available when using the “Use Customized Message” option. It lets you to set the font color.

Variable for Table

It allows to use a variable to change a part of the string containing the table name.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the table you want to replace (replace from).

Fields Variables

It allows to use a variable to change a part of the string containing the field name.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the field you want to replace (replace from).

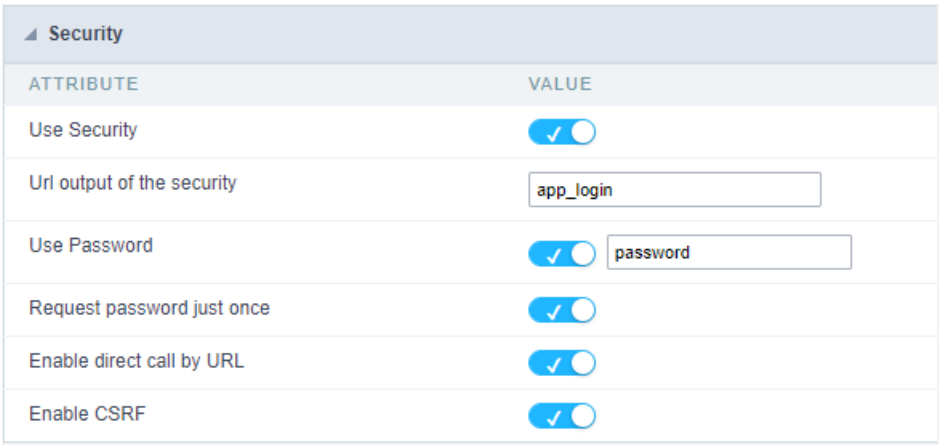
Case sensitive

It defines if the database connection uses case sensitive or not.

Related Links 🔗

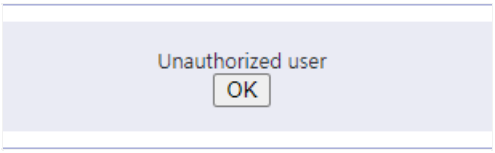
Related Videos ▷

Security



Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application.



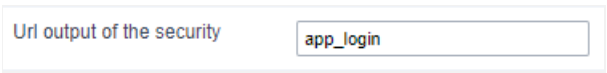
When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

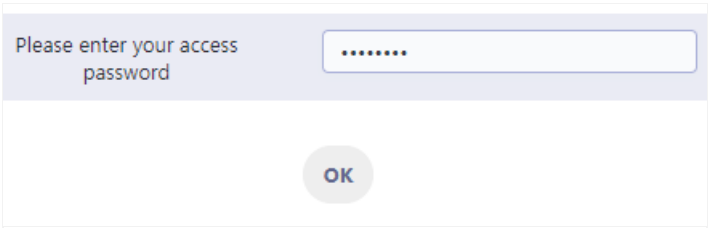
Defines which application the user will be redirected to after the “unauthorized user” warning.



Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.



The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

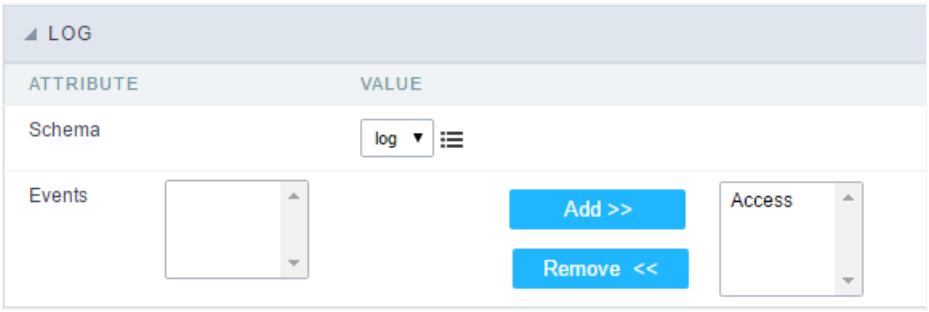
Related Links

Related Videos

PDF Report Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

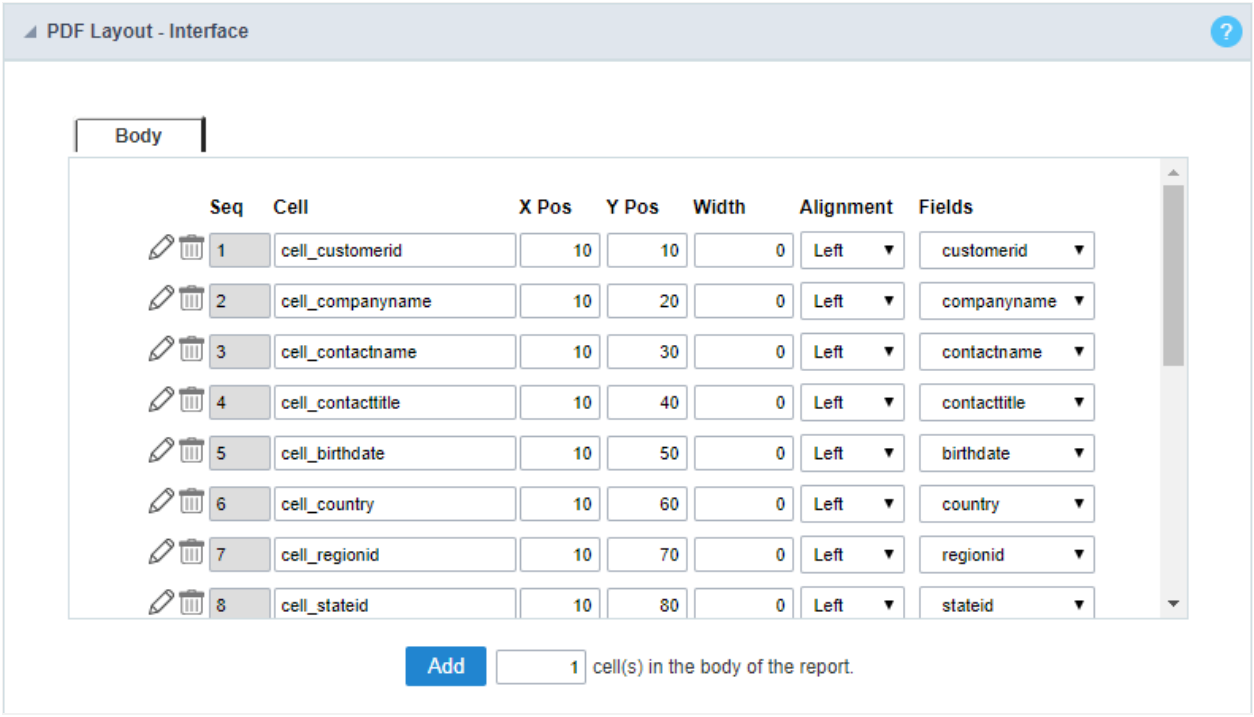
Related Links

Related Videos

PDF Advanced settings

Interface

Using this interface you can define the display layout of cells containing the values of the fields in the body of the report.



PDF interface settings.

- **Attributes**
  - **Seq** : Sequential number of cells (ascending order).
  - **Cell** : Cell name.
  - **Pos X** : Sets the abscissa of the cell.
  - **Pos Y** : Sets the ordinate of the cell.
  - **Width** : Sets the cell width.
  - **Alignment** : Sets the cell alignment.
  - **Field** : Here you have to select the field that will be displayed within the cell, according to the application SQL.

Code

Scriptcase creates the codes automatically, so by changing that within the option “code” you will assume the PDF creation PHP code.

Configuration interface of the PDF Code.

The application “Rport PDF” was developed based on a library called TCPDF, so that you can use the available library methods or a corresponding macro from the table below. In order to use some method you must use $pdf object, for example: $pdf->AcceptPageBreak(parameters).

In order to access the TCPDF documentation Click here

Macros ReportPDF

Scriptcase Macro	Description
sc_pdf_text	This method allows you to place a string.
sc_pdf_write	This method prints the text of the current position.
sc_pdf_set_y	Sets the current y-axis position.
sc_pdf_set_xy	Sets the current position of the x-axis and y-axis.
sc_pdf_set_top_margin	Defines the top of the margin.
sc_pdf_set_right_margin	Sets the right margin.
sc_pdf_set_left_margin	Sets the left margin.
sc_pdf_set_margins	Sets the left, top, and right margin.
sc_pdf_set_title	Defines the title of the document.
sc_pdf_set_text_color	Sets the color to use for the text.
sc_pdf_set_subject	Defines the subject of the document.
sc_pdf_set_line_width	Sets the length of the pdf line.

Scriptcase Macro	Description
sc_pdf_line	Draws a line between two points.
sc_pdf_set_keywords	Associates keywords with the document.
sc_pdf_set_font	Sets the current font size.
sc_pdf_add_font	Defines which font will be used in the text.
sc_pdf_close	Defines the closing of the pdf document.
sc_pdf_error	This method is automatically called in case of a fatal error.
sc_pdf_add_page	Adds a new page to the pdf document.
sc_pdf_footer	This method is responsible for rendering the footer of the pdf document.
sc_pdf_header	This method is responsible for rendering the header of the pdf document.
sc_pdf_get_string_length	Returns the length of a string.
sc_pdf_get_y	Returns the current y-axis position.
sc_pdf_get_x	Returns the current x-axis position.
sc_pdf_link	Adds the link to a rectangular area of the page.
sc_pdf_image	Embed an image in the document.
sc_pdf_set_author	Defines the author of the document.
sc_pdf_set_auto_page_break	Enables or disables the automatic page breaking mode.
sc_pdf_set_compression	Activates or deactivates page compression.
sc_pdf_output	Send the document to a given destination.
sc_pdf_set_creator	Defines the creator of the document.
sc_pdf_set_display_mode	Defines the way the document is to be displayed by the viewer.
sc_pdf_accept_page_break	Whenever a page break condition is met, the method is called.
sc_pdf_ln	Performs a line break.
sc_pdf_rect	Outputs a rectangle. It can be drawn (border only), filled (with no border) or both.
sc_pdf_set_draw_color	Defines the color used for all drawing operations (lines, rectangles and cell borders)
sc_pdf_set_fill_color	Defines the color used for all filling operations.
sc_pdf_multi_cell	This method allows printing text with line breaks.
sc_pdf_cell	Prints a cell (rectangular area) with optional borders, background color and character string.

Scriptcase Macro	Description
sc_pdf_add_link	Creates a new internal link and returns its identifier.
sc_pdf_alias_nb_pages	Defines an alias for the total number of pages.
sc_pdf_set_link	Defines the page and position a link points to.

sc_pdf_text

Description:

This method allows to place a string precisely on the page.

Parameters:

y: y-axis
txt: string
link: URL

Example:

```
sc_pdf_text("1","1","Your Text Here");
```

Scope:

Layout PDf **>** *Código* **>** *Layout e Corpo*

sc_pdf_write

Description:

This method prints text from the current position. When the right margin is reached (or the \n character is met) a line break occurs and text continues from the left margin. Upon method exit, the current position is left just at the end of the text. It is possible to put a link on the text.

Parameters:

h: height
txt: string
link: URL

Example:

```
sc_pdf_write(150,'Your Text Here','https://www.scriptcase.net');
```

Scope: *PDf Layout > Code > Layout & body*

sc_pdf_set_y

Description:

Sets the current y-axis position.

Parameters:

y: y-axis

Example:

```
sc_pdf_set_y(180,true);
```

Scope: *PDf Layout > Code > Layout & body*

sc_pdf_set_x

Description:

Sets the current x-axis position.

Parameters:

x: x-axis

Example:

```
sc_pdf_set_x(180);
```

Scope: *PDf Layout > Code > Layout & body*

sc_pdf_set_xy

Description:

Sets the current position of the x-axis and y-axis.

Parameters:

x: y-axis
y: x-axis

Example:

```
sc_pdf_set_x(180,180);
```

Scope: *PDf Layout > Code > Layout & body*

sc_pdf_set_top_margin

Description:

Defines the top margin. The method can be called before creating the first page.

Parameters:

margin: the margin

Example:

```
sc_pdf_set_top_margin(1000);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_right_margin

Description:

Defines the right margin. The method can be called before creating the first page.

Example:

```
sc_pdf_set_right_margin(1000);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_left_margin

Description:

Defines the left margin. The method can be called before creating the first page.

Example:

```
sc_pdf_set_left_margin(1000);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_margins

Description:

Defines the left, top and right margins. By default, they equal 1 cm. Call this method to change them.

Parameters:

left: margin left
top: margin top
right: margin right

Example:

```
sc_pdf_set_margins(200, 200, 200);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_title

Description:

Defines the title of the document.

Parâmetro:

`título: title of the document`

Example:

```
sc_pdf_set_title("Your Title Here");
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_text_color

Description:

Defines the color used for text. It can be expressed in RGB components or gray scale. The method can be called before the first page is created and the value is retained from page to page.

Example:

```
sc_pdf_set_text_color(106,13,173);
sc_pdf_write(150,'Your Text Here','https://www.scriptcase.net');
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_subject

Description:

Defines the subject of the document.

Example:

```
sc_pdf_set_subject("Your Subject here");
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_line_width

Description:

Defines the line width.

Parameters:

width: margin width

Example:

```
sc_pdf_set_line_width(1500);
sc_pdf_line(50,25,70,35);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_line

Description:

Draws a line between two points.

Example:

```
sc_pdf_line(50,25,70,35);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_keywords

Description:

Associates keywords with the document, generally in the form ‘keyword1 keyword2 …’.

Parameters:

keywords: list of keywords

Example:

```
sc_pdf_set_keywords(test test2, true);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_font

Description:

Sets the font used to print character strings. It is mandatory to call this method at least once before printing text or the resulting document would not be valid.

Parameters:

family

Font to be defined.

- Courier (fixed-width)
- Helvetica or Arial (synonymous; sans serif)
- Times (serif)
- Symbol (symbolic)
- ZapfDingbats (symbolic)

style

Font style to be set.

- empty string: **regular**
- B: **bold**
- I: **italic**
- U: **underline**

size

Font size to be set.

Example:

```
sc_pdf_set_font('Arial', 'B', 14)
```

Scope: *PDf Layout > Code > Layout & body*

sc_pdf_add_font

Description:

Defines which font will be used in the text. You can pass parameters for font-family, font-style and font-size.

Parameters:

family

Font to be defined.

- Courier (fixed-width)
- Helvetica or Arial (synonymous; sans serif)
- Times (serif)
- Symbol (symbolic)
- ZapfDingbats (symbolic)

style

Font style to be set.

- empty string: **regular**
- B: **bold**
- I: **italic**
- U: **underline**

size

Font size to be set.

file

The font file

By default, the name is constructed from the family and style, in lower case with no spaces.

Example:

```
sc_pdf_add_font('Comic', 'I', 'comici.php');
```

Scope:

Layout PDf > Code > Layout & body

sc_pdf_close

Description:

Terminates the PDF document. It is not necessary to call this method explicitly because Output() does it automatically.

Example:

```
sc_pdf_close();
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_error

Description:

This method is automatically called in case of a fatal error; it simply throws an exception with the provided message.

Example:

sc_pdf_error(“Your Text Here”);

Scope: *PDf Layout > Code > Layout & body*

sc_pdf_add_page

Description:

Adds a new page to the document.

Example:

sc_pdf_add_page('Portrait','A4',0);

Scope:

PDf Layout > Code > Layout & body

sc_pdf_footer

Description:

This method is used to render the page footer. It is automatically called by sc_pdf_add_page() and sc_pdf_close() and should not be called directly by the application.

Example:

sc_pdf_footer();

Scope:

PDf Layout > Code > Layout & body

sc_pdf_header

Description:

This method is used to render the page header. It is automatically called by sc_pdf_add_page() and should not be called directly by the application.

Example:

```
sc_pdf_header();
```

Scope: *PDf Layout > Code > Layout & body*

sc_pdf_get_string_length

Description:

Returns the length of a string in user unit. A font must be selected.

Example:

```
sc_pdf_get_string_length("Text To Be Measured");
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_get_y

Description:

Returns the current y-axis position.

Example:

```
sc_pdf_get_y();
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_get_x

Description:

Returns the current x-axis position.

Example:

sc_pdf_get_x();

Scope:

PDf Layout > Code > Layout & body

sc_pdf_link

Description:

Puts a link on a rectangular area of the page. Text or image links are generally put via Cell(), Write() or Image(), but this method can be useful for instance to define a clickable area inside an image.

Parameters:

x : x-axis
y : y-axis
w : width
h : height
link : url

Example:

sc_pdf_link(100,100,10,10,'https://scriptcase.com.br');

Scope:

PDf Layout > Code > Layout & body

sc_pdf_image

Description:

Embeds an image in the document. Image size can be set in several ways. Supported formats: JPEG, PNG and GIF, the GD extension must be enabled to use GIF.

Parameters:

link : url path
x : x-axis
y : y-axis
w : width
h : height
type : image format

Example:

sc_pdf_image('http://chart.googleapis.com/chart?cht=p3&chd=t:60,40&chs=250x100&chl=HelloWorld', 60, 30, 90, 0,'PNG');

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_author

Description:

Defines the author of the document.

Example:

```
sc_pdf_set_author("Scriptcase");
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_auto_page_break

Description:

Enables or disables the automatic page breaking mode. When enabling, the second parameter is the distance from the bottom of the page that defines the triggering limit. By default, the mode is on and the margin is 2 cm.

Example:

```
sc_pdf_set_auto_page_break(true, 10);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_compression

Description:

Activates or deactivates page compression. When activated, the internal representation of each page is compressed, which leads to a compression ratio of about 2 for the resulting document.

Note: the **Zlib** extension is required for this feature. If not present, compression will be turned off.

Example:

```
sc_pdf_set_compression(true);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_output

Description:

Send the document to a given destination: browser, file or string. In the case of a browser, the PDF viewer may be used or a download may be forced.

Parameters:

- I: **send the file inline to the browser. The PDF viewer is used if available.**
- D: **send to the browser and force a file download with the name given by name.**
- F: **save to a local file with the name given by name (may include a path).**
- S: **return the document as a string.**

Example 1: Save the document to a local directory:

```
$pdf->Output('F', 'reports/report.pdf');
```

Example 2: Force a download:

```
$pdf->Output('D', 'report.pdf');
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_creator

Description:

Defines the creator of the document. This is typically the name of the application that generates the PDF.

Example:

```
sc_pdf_set_creator(“X Creator”,”true”);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_display_mode

Description:

Defines the way the document is to be displayed by the viewer.

Parameters:

zoom:

- *fullpage:* displays the entire page on screen
- *fullwidth:* uses maximum width of window

- *real:* uses real size (equivalent to 100% zoom)
- *default:* uses viewer default mode

layout:

- *single:* displays one page at once
- *continuous:* displays pages continuously
- *two:* displays two pages on two columns
- *default:* uses viewer default mode

Example:

```
sc_pdf_set_display_mode("fullpage", "");
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_accept_page_break

Description:

Whenever a page break condition is met, the method is called, and the break is issued or not depending on the returned value. This method is called automatically and should not be called directly by the application.

Example:

```
sc_pdf_accept_page_break();
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_ln

Description:

Performs a line break. The current abscissa goes back to the left margin and the ordinate increases by the amount passed in parameter.

Example:

```
sc_pdf_ln();
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_rect

Description:

Outputs a rectangle. It can be drawn (border only), filled (with no border) or both.

Parameters:

- *x* : x-axis
- *y* : y-axis
- *w* : width
- *h* : height

Estilo de renderização:

D or empty string: draw. This is the default value. F: fill DF ou FD: Draw and fill

Example:

```
sc_pdf_rect($x, $y, $w, $h, 'D');
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_draw_color

Description:

Defines the color used for all drawing operations (lines, rectangles and cell borders). It can be expressed in RGB components or gray scale. The method can be called before the first page is created and the value is retained from page to page.

Example:

```
sc_pdf_set_draw_color(245, 232, 125);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_fill_color

Description:

Defines the color used for all filling operations (filled rectangles and cell backgrounds). It can be expressed in RGB components or gray scale. The method can be called before the first page is created and the value is retained from page to page.

Example:

```
sc_pdf_set_fill_color(245, 232, 125);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_multi_cell

Description:

This method allows printing text with line breaks. They can be automatic (as soon as the text reaches the right border of the cell) or explicit (via the \n character). As many cells as necessary are output, one below the other.

Parameters:

- ***w:*** cell width
- ***h:*** cell height
- ***txt:*** string to print
- ***border:*** Indicates if borders must be drawn around the cell block.
- ***align:*** Sets the text alignment. Possible values are:
  L: left alignment
  C: center
  R: right alignment
  J: justification (default value)
  fill: Indicates if the cell background must be painted (true) or transparent (false). Default value: false.

Example:

```
_sc_pdf_multi_cell(63, 10, "YourText", 1);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_cell

Description:

Prints a cell (rectangular area) with optional borders, background color and character string. The upper-left corner of the cell corresponds to the current position. The text can be aligned or centered. After the call, the current position moves to the right or to the next line. It is possible to put a link on the text. If automatic page breaking is enabled and the cell goes beyond the limit, a page break is done before outputting.

Parameters:

- ***w:*** cell width
- ***h:*** cell height
- ***txt:*** string to print
- ***border:*** Indicates if borders must be drawn around the cell block.
- ***align:*** Sets the text alignment. Possible values are:
  L: left alignment
  C: center
  R: right alignment
  J: justification (default value)
  fill: Indicates if the cell background must be painted (true) or transparent (false). Default value: false.
- ***link:*** URL

Example:

```
sc_pdf_cell(60, 6, 'Your Text', 0);
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_add_link

Description:

Creates a new internal link and returns its identifier. An internal link is a clickable area which directs to another place within the document.

Example:

```
sc_pdf_add_link();
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_alias_nb_pages

Description:

Defines an alias for the total number of pages. It will be substituted as the document is closed.

Example:

```
sc_pdf_cell(0, 10, 'Page','{nb}', 0, 0, 'C')
sc_pdf_alias_nb_pages();
```

Scope:

PDf Layout > Code > Layout & body

sc_pdf_set_link

Description:

Defines the page and position a link points to.

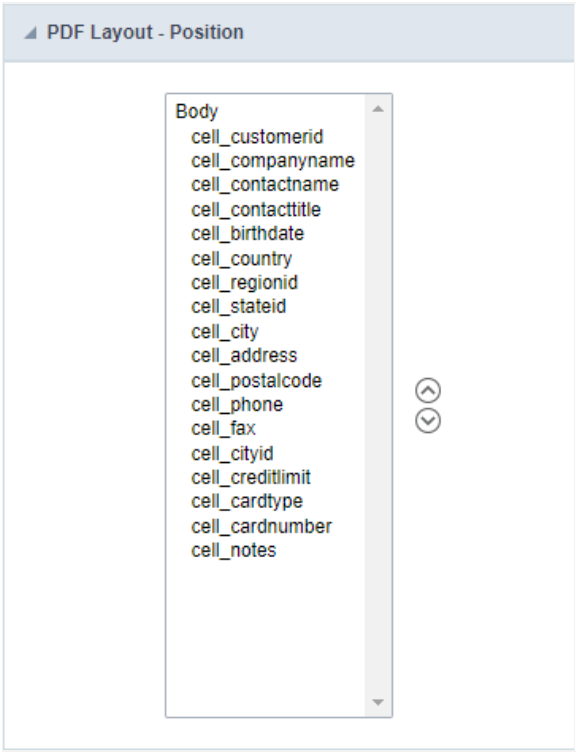
Example:

```
sc_pdf_set_link($link);
```

Scope:

Positioning

This interface allows you to define the display order of the fields (selecting through the arrows next to the right frame).



Configuration interface of the PDF fields placement.

Related Links

Related Videos

PDF Report - Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid	✓	✓
Procedure	✓	✓
Chart	✓	✓
Form	✓	✓
Control	✓	✓
Calendar	✓	✓
Search	✓	✓
PDF Report	✓	✓

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven't manage fields.

Related Links

Related Videos ▷

- [d template]
- [Invoice template]

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Text Field

General Settings

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  ```
  SELECT display_field FROM table WHERE key_field = {grid_field}
  ```

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the "Separated by" field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.

- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).

- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Multiple Lines Text Field

General Settings

Multiple Lines Text field Configuration Interface.

- **Data Type** : Define the type of field for the application. When it is defined as a Multiple Lines Text, it accepts letters, numbers and special characters in multiple lines.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Case Settings** : Convert the letter from the field when losing focus. The options are:
  - **Upper case** : All in Upper Case
  - **Lower case** : All in Lower case
  - **Capitalize first word** : Capitalizes the first letter of the first word
  - **Capitalize all words** : Capitalizes the first letter of all the the words
- **Show HTML content** : Determines if the HTML contained in the field will be displayed or not. If enabled, the HTML will be displayed, otherwise the HTML will be interpreted by the browser.
- **Grid Mask** : Defines the mask for the field display.There are two typed of masks described below:

Character	Description
X	Placeholder to any character. Replaced by any character. If number of characters entered are less then the mask size, the field value is completed with zeros (Filling full size field entry is required).
Z	Replaced by any character retrieved from database. Suppress zeros at field left (Complete field filling is optional). When used combined with the mask character X it should be placed at the mask left.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

- **Do Not Repeat Value** : Do not repeat the value of the field in case it is the same as the previous record.

- **Run content in JavaScript** : If enabled, the JavaScript will be interpreted by the browser, otherwise the JavaScript will be displayed.
- **SQL Type** : Informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  ```
  SELECT display_field FROM table WHERE key_field = {grid_field}
  ```

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the "Separated by" field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial

Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links

- Business card template
- [illegible]plate

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Integer Field

On this page, the developer can configure the settings for your field of type number. From the use of specific display symbols to the mode in which they are displayed. Thus, it can streamline its application.

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Color for the negative values:

It allows you to define a color when the value is negative, improving the understanding of the end user about that kind of value.

Example:

Display the value in words:

The value of the field will be displayed in full on application. This feature can facilitate the comprehension and understanding of the user.

Example:

Line size:

Maximum size in characters to be displayed in the value cell, in full. When this value is exceeded the line will break within the cell.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:

Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

```
SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}
```

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture

Assigned value	Description in Lookup
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Related Links 🔗

- Business card template
- Invoice template

Related Videos ▷

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Decimal Field

General Settings

Decimal field Configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

 Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the

select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.

- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links 🔗

- Business card template
- Invoice template

Related Videos ▷

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Percent Field

General Settings

Percentage field Configuration Interface.

Data Type

Define the field type to Percentage.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value

returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and

lookup value is enabled.

Related Links 🔗

- Business card template
- Invoice template

Related Videos ▷

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Currency Field

General Settings

Currency field Configuration Interface.

Data Type

Define the field type to Currency. It allows defining the currency number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	Placeholder to any character. Replaced by any character. If number of characters entered are less then the mask size, the field value is completed with zeros (Filling full size field entry is required).
Z	Replaced by any character retrieved from database. Suppress zeros at field left (Complete field filling is optional). When used combined with the mask character X it should be placed at the mask left.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

![Displaying repeated values.][cons_configuracao_geral_texto_nao_repetir_valor]

SQL Type

It informs the data type of field in the database.

Values Format

Interface of Values Format.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Currency Format

It defines if the field displays the Currency Symbol of the Regional Settings.

Currency Symbol

It allows setting the character that represents the Currency Symbol.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Display full value

It displays the full value. Example: 2018(Two thousand eighteen).

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the "Separated by" field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.

- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - HTML Image Field

General Settings

HTML Image field Configuration Interface.

Data Type

Define the type of field. When setting it to HTML Image, it allows to display an image into the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Image

Define an image to be displayed. The icon “Select Image” lists all images from Scriptcase and your uploaded images. The “Upload an image” option allows you to send a copy to the Scriptcase server.

New Link

This button allows the creation of a field link with some other application..

This way, it is possible, for example, to create a link with a blank to delete a record from the grid, passing the record ID as a parameter.

> See more information about field binding by clicking here.

Border

Define the width of the Image border in Pixels.

Width

Define the image width size in Pixels.

Height

Define the image height size in Pixels.

Related Links

- Business card template
- [illegible]plate

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Credit Card Field

General Settings

Credit Card Number Configuration Interface.

- **Data Type** : Select the type of field for the application. When it is defined as a Credit Card Number, you can define some rules for the display format of the Credit Card.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **Do Not Repeat Value** : Do not repeat the value of the field in the case it is the same as the previous record.
- **SQL Type** : Informs the data type of the field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  ```
  SELECT display_field FROM table WHERE key_field = {grid_field}
  ```

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains

this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links

- Business card template
- [illegible]plate

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - E-mail Field

General Settings

Email field Configuration Interface.

- **Data Type** : Defines the type of field for the application. When it is defined as an Email, when you click on the field you be offered a choice for your email client and send an email to that specific email.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Do Not Repeat Value** : Do not repeat the value of the field in case it is the same as the previous record.
- **SQL Type** : Informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links

- Business card template
- Invoice template

Related Videos

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - URL Field

General Settings

URL field Configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  ```
  SELECT display_field FROM table WHERE key_field = {grid_field}
  ```

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Date Field

General Settings

Date field Configuration Interface.

Data Type

Define the type of field. When setting it to Date, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

 Format of Values without Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying dates.

Detail Mask

Define the format of the field in the Grid. It uses the patterns of PHP function Date.

d-m-Y	25-09-2009
F/Y	September/2009
j/n/Y \a\s g:i:s A	25/9/2001 as 14:30:11 PM
l, d \d\e F \d\e Y	Thursday, 25 of January of 2009
h:i:s	11:33:20
#h:i:s	123:43:27 (accumulating the hours)

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

- **Example 1** : If the field is storing the Date value in a **char(8)** where the positions **1** to **4** represents the year, the positions **5** and **6** is the month and the position **7** and **8** is the day, you must define the internal format as: **YYYYMMDD**
- **Example 2** : If the field is storing the Date value in a **char(6)** where the positions **1** and **2** represents the month and the positions **3** to **6** is the year, you must define the internal format as: **MMYYYY**

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the "Separated by" field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).

- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure

Attribute Value	Lookup Description
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Time Field

General Settings

Time field Configuration Interface.

Data Type

Define the type of field. When setting it to Time, you can inform a time format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

 Format of Values without Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying dates.

Detail Mask

Define the format of the field in the Grid. It uses the patterns of PHP function Date.

d-m-Y	25-09-2009
F/Y	September/2009
j/n/Y \a\s g:i:s A	25/9/2001 as 14:30:11 PM
l, d \d\e F \d\e Y	Thursday, 25 of January of 2009
h:i:s	11:33:20
#h:i:s	123:43:27 (accumulating the hours)

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

- **Example 1** : If the field is storing the Date value in a **char(8)** where the positions **1** to **4** represents the year, the positions **5** and **6** is the month and the position **7** and **8** is the day, you must define the internal format as: **YYYYMMDD**
- **Example 2** : If the field is storing the Date value in a **char(6)** where the positions **1** and **2** represents the month and the positions **3** to **6** is the year, you must define the internal format as: **MMYYYY**

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the "Separated by" field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).

- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure

Attribute Value	Lookup Description
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Date and time Field

General Settings

Date and Time field Configuration Interface.

Data Type

Define the type of field. When setting it to Date and Time, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

 Format of Values without Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying dates.

Detail Mask

Define the format of the field in the Grid. It uses the patterns of PHP function Date.

d-m-Y	25-09-2009
F/Y	September/2009
j/n/Y \a\s g:i:s A	25/9/2001 as 14:30:11 PM
l, d \d\e F \d\e Y	Thursday, 25 of January of 2009
h:i:s	11:33:20
#h:i:s	123:43:27 (accumulating the hours)

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

- **Example 1** : If the field is storing the Date value in a **char(8)** where the positions **1** to **4** represents the year, the positions **5** and **6** is the month and the position **7** and **8** is the day, you must define the internal format as: **YYYYMMDD**
- **Example 2** : If the field is storing the Date value in a **char(6)** where the positions **1** and **2** represents the month and the positions **3** to **6** is the year, you must define the internal format as: **MMYYYY**

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the "Separated by" field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).

- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure

Attribute Value	Lookup Description
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Image (Database) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Image (Database), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Image Border

Width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Open in Another Window

Allows to open the image in another window.

SQL Type

It informs the data type of field in the database.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Image (File Name) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Image (File Name), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Image Border

Width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API.

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API.

Open in Another Window

Allows to open the image in another window.

Subdirectory for local storage

Local subdirectory or on the server where the files are stored. eg: {CustomerId} , [glo_var_seq]

Image Caching

Time in minutes the image cache will be kept in the server before being deleted.

Repeat value

Repeat the field value if it is equal to the previous record.

SQL Type

It informs the data type of field in the database.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Document (Database) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Document (Database), all the document files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Icon

Displays an icon beside the field to identify the type of document.

File Name

It allows defining the field to store the name of the document in the database.

SQL Type

It informs the data type of field in the database.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Document (File Name) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Document (File Name), all the document files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Sub-folder

Sub-folder name that the files are stored.

Icon

Displays an icon beside the field to identify the type of document.

File Name

It allows defining the field to store the name of the document in the database.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API.

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API.

SQL Type

It informs the data type of field in the database.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - QRCode Field

General Settings

QRCODE field Configuration Interface.

Data Type

Define the type of field. When setting it to QRCODE, it allows you to transform values into a QRCODE.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

SQL Type

It informs the data type of field in the database.

Watch below a video showing an example with QRCODE

Values Format

Interface of Values Format.

Level of error correction

The Codewords are 8 bits long and use the Reed–Solomon error correction algorithm with four error correction levels. The higher the error correction level, the less storage capacity.

Image Size

Set the size of the QRCODE image.

Margin

Set the margins of the QRCODE.

 Interface of Values Format.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.

- **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
- **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
- **Background Image** : Allows you to add a background image to the field.
- **Line Break** : Enables the possibility of a line break for the field.
- **Content color** : Changes the content color of the query field and the grouped field, when available.
- **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Related Links 🔗

- Business card template
- Invoice template

Related Videos ▷

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - Bar code Field

General Settings

Configuration Interface of the Barcode Field.

- **Data Type** : DataType of the field for the application.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **SQL Type** : Database field type.

Watch below a video showing an example about the Barcode field:

Values Format

Configuration Interface of the Barcode Field.

- **Type** : Type of Barcode.
- **Text** : Barcode Text for illustration purposes.
- There are **18 types of barcodes**, that are listed below:

Barcode configuration interface.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.

- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Related Links 🔗

- Business card template
- Invoice template

Related Videos ▷

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report - SubSelect Field

General Settings

SubSelect field configuration Interface.

Increment : Defines the space between the record lines of the SubSelect field. **Connection** : Displays a list of connections created in the current project. **Table** : Displays the tables related to the database connection selected previously. **SubSelect** : Displays the SQL of the chosen table in the previous option. Allowed to be modified and use field variables and global variables. **SQLBuilder** : Opens the ScriptCase SQLBuilder, to build the SQL commands.

Related Links

- Business card template
- [illegible]plate

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

Signature

Configuración General

Configuration Interface of the Signature Field.

The signature field will help you creating more sofisticated forms and making it possible to store signatures in your database. Inside our development environment we have specific settings that will help you to customize your field, those options are:

- **Data Type** : You can define the type of field for the application. When it is defined as a text, it accepts letters, numbers and special characters.
- **Label** : Lets you define a label to the field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Background color** : Defines a color to the field background by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Pen color**: Set a color to the pen by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Width** : Lets you define a width to the field.
- **Height** : Set a height to the field.
- **Subtitle** : Defines the subtitle that will be displayed beside the field.
- **Initial Value** : Lets you define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
- **Save Variable** : Save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Disabled Field** : Define if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : Displays the HTML Object that will be used to display the field in the form.
- **SQL Type** : Displays the data type of field in the database.

Configuración de Visualización

You can set CSS values for the field individually. Thus, for example, you can highlight one field of the others in the application. As CSS field properties, when changed, they are added to a class created automatically by ScriptCase for each application field.

> Individual field CSS settings, when inserted, override theme settings (CSS of applications (Themes)) selected for application.

The settings are divided into three property blocks, these blocks are:

CSS of Title

This makes it possible to change the CSS properties of the field’s Label.

In the example below, You can see the difference of field title configurations. While the field **Contactname** have the same formatting, inherits the theme of the application, the field **Contacttitle** have a different formation of of others, from the changes made in the CSS of the field.

CSS of field

Changes the CSS properties of the <td> where the input object (where the user enters data for insertion into a form) is positioned. In the image below, you can see where the change is applied.

Field Contacttitle with changes to Field CSS properties, changing background color and horizontal alignment

CSS of Input object

Changes the CSS properties in the Input of field object, where the user type the data in a form.

Field with changes to CSS properties of Input Object, changing background color and input source color

CSS properties

The available configuration options are basically the same for each of the configuration blocks above.

description of available configuration attributes

Font

Changes the font of the text according to the fonts selected using the *font-family* property.

In this option, some types of fonts are provided to you (as shown below).

Font Size

Changes the font size of the text using the property *font-size* in the field class.

You need to select the available value from our list, the measure used for this property is the pixel.

Font Color

Define the font color used by the property *color* in the field class.

Background color

Define the background color using the property *background-color* in the field class.

The colors that will be used in the two color properties listed above, **Font color** and **Background color**, can be entered via the color palette - - available next to the field or manually entered values in the accepted formats that are: *Hexadecimal*, *RGB*, *RGBA*, *HSL*, *HSLA* or o *Color name*.

Color palette
By clicking on the color palette icon - - next to the field, a window will open with some default colors.

When you select one of the colors, a value in hexadecimal format (HEX) will be entered, representing the chosen color.

Hexadecimal
Acronym for hexadecimal, this code is composed of the pound sign (#) plus six digits. The first two define the intensity of the color red, the middle two are green and the last two are blue.

Bold

Applies bold style to the font.

Underline

Lets you apply the underlined style to the font.

Border style

Defines the border font style.

Border Collapse

Defines the border collapse.

Border size

Changes the size of the title border.

Border color

Chooses the border color , using a color palette to apply to the title.

Horizontal alignment

Position the filter label at desired location (left,rigth,center e justify).

Vertical alignment

Position the label of the filter in the wanted location baseline, sub, super, top, text-top, middle, bottom, text-bottom).

> Not available in Input Object CSS

Width

To set the width of the title.

Height

To set the height of the title.

Related Links

- Business card template
- Invoice template

Related Videos

- Report PDF overview
- PDF reports with the exports option
- PDF Viewer Part 1/2
- PDF Viewer Part 2/2
- Macro sc_set_pdf_name
- PDF Authentication Variables

PDF Report Events Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application's CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event's PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Report PDF

This event occurred before the application execute the SQL, and execute only once. Is used to do verification of variables, and security verification.

onScriptInit - Report PDF

This event occurs when the application its loaded, or reload. Occurs before the application runs the SQL statement, so using this event it is possible to modify the grid SQL statement dynamically, based on any logical. Macros frequently called on this event: sc_select_field, sc_select_order, sc_select_where(add), etc

onRecord - Report PDF

This event occurs immediately before displaying a row, it is used to calculate any cell value.

onHeader - Report PDF

This event occurs before the header display. This event is frequently used when it is required to display any calculated value on report header.

onFooter - Report PDF

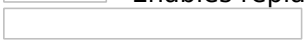
This event is used when in the footer we need to display some calculated value, it is in this event that we write the calculation necessary to assemble some value that we want to display.

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

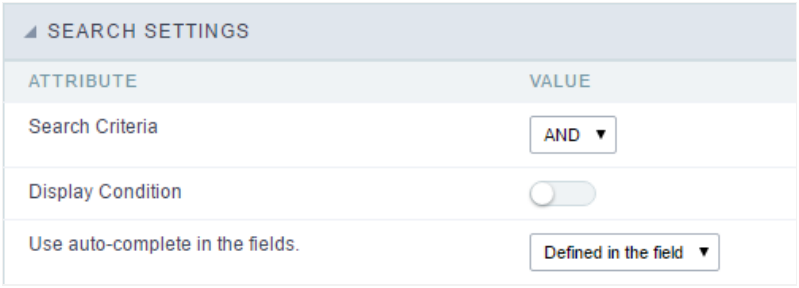
The hotkeys are available in all browsers compatible with Scriptcase.

> Click Here to view the Scriptcase hotkeys documentation.

Search

Settings

With this interface, you can define general options of the Search Form.



Search configuration Interface.

Search Criteria

Allows to select the logical operator **AND** or **OR** to define the criteria of the search;

Display Condition

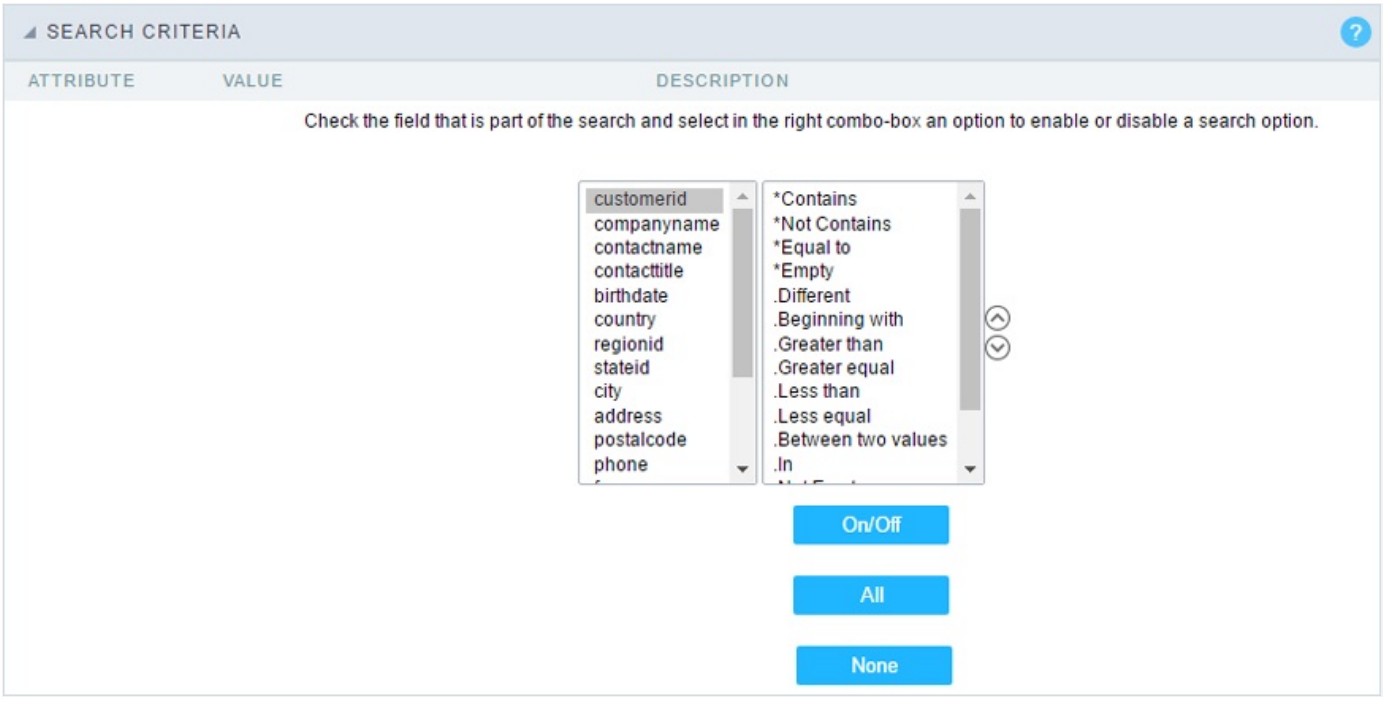
Gets the condition of the search available for the user to choose one. He can select “AND” or “OR” in a Combobox.

Use auto-complete in the fields

Automatically turns the field into an autocomplete according to the existing values in the database. If the user chooses **Yes**, the autocomplete will enable automatically in all inputs that contain a relationship. If the user decides **No**, so no autocompletes will be displayed. Otherwise, the option selected is **Defined in the field** it’ll keep the settings for each field individually.

Search Criteria

With this interface, you can configure the conditions available for each field of the Search form.



Search configuration Interface.

We can see the fields list on the left combo. On the right, the list of options for filtering the selected field. To select an option, click on one of them (Equal to, Beginning with, Contains, etc.) and then the button On/Off. The arrows, on the right, allows altering the order of the fields.

For the Date type fields, you can define special conditions for the search, accessing the field configurations, and editing the Special Conditions Settings.

Below the list are the buttons to enable the selected options:

- **On/Off**: Enables or disables the field or the option chosen.
- **All**: Marks all fields or options.
- **None**: Unmarks all the fields or options.

Advanced Search

Settings

Through the table below we are able to set all the options that will be part of the application “Grid Search”.

Margins

Defines the position of the margins of the Search Form.

Keep Values

It keeps the searched values when the user returns to the search form.

Keep Columns and Order Selection

Set it to preserve the selected columns and sorting for each search, if they went changed by the user through the toolbar options.

Use Enter to

It allows you to define the action that the Enter Key has on the Search form.**Tabulate** enables you to navigate between fields, and **Submit** performs the search(activates the Search button).

Display Tags

Allows displaying as tags, the searches used for the Grid.

Display after filtering

Display tags only after performing an advanced search. If disabled, it will always display a tag, regardless of the advanced search.

Unify results

Sets the chars limit to group the result of the tags. This option should be used when the field type is multiple-select.

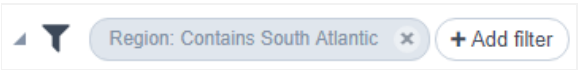
Treeview in the Tags

Sets the use of Treeview for tags.

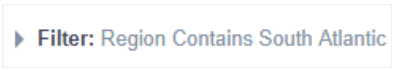
Initial status of the Treeview

It sets the initial state of the Treeview. (If the app is using Treeview for tags)

Start open

It displays the tags. 

Start close

It displays the full description of the tags as text. 

Clear other filters after submit

After applying the advanced filter, all other filters will be deleted.

Interact with the dynamic filter

Apply the same filter in matching fields between Advanced and Dynamic Filter, when not showing the labels.

Highlight

Highlights the results in the query. It only works for “Exactly the same”, “Contains” and “Equal start” conditions.

To customize how the highlight will be displayed, you must access the theme editing tool.

1 - Access the **Layout > Application Themes** menu.
2 - Choose the desired theme and click **Advanced Mode**.
3 - Then look for the sub-item **Grid**, within the item of the same name.
4 - In this sub-item you will find the option “Line”, which contains the folders **Line Odd and Even**. In each of these folders you will find the option **Quicksearch highlight**, with the options for editing Text, Border and Background of the searched terms.

Select fields

![Interface for filter fields selection.][filtro_avancado_configuracao_selecionar_campos] *Interface for filter fields selection.*

Required

Defines which fields of application will be required for the Search.

![Required fields interface.][filtro_avancado_configuracao_campos_obrigatorios] *Required fields interface.*

The application generated will be displayed a bullet (*) next to the field and an error message is generated if not assigned no value.

![Configuration interface of the marker placement.][filtro_avancado_configuracao_campos_obrigatorios_posicionamento] *Configuration interface of the marker placement.*

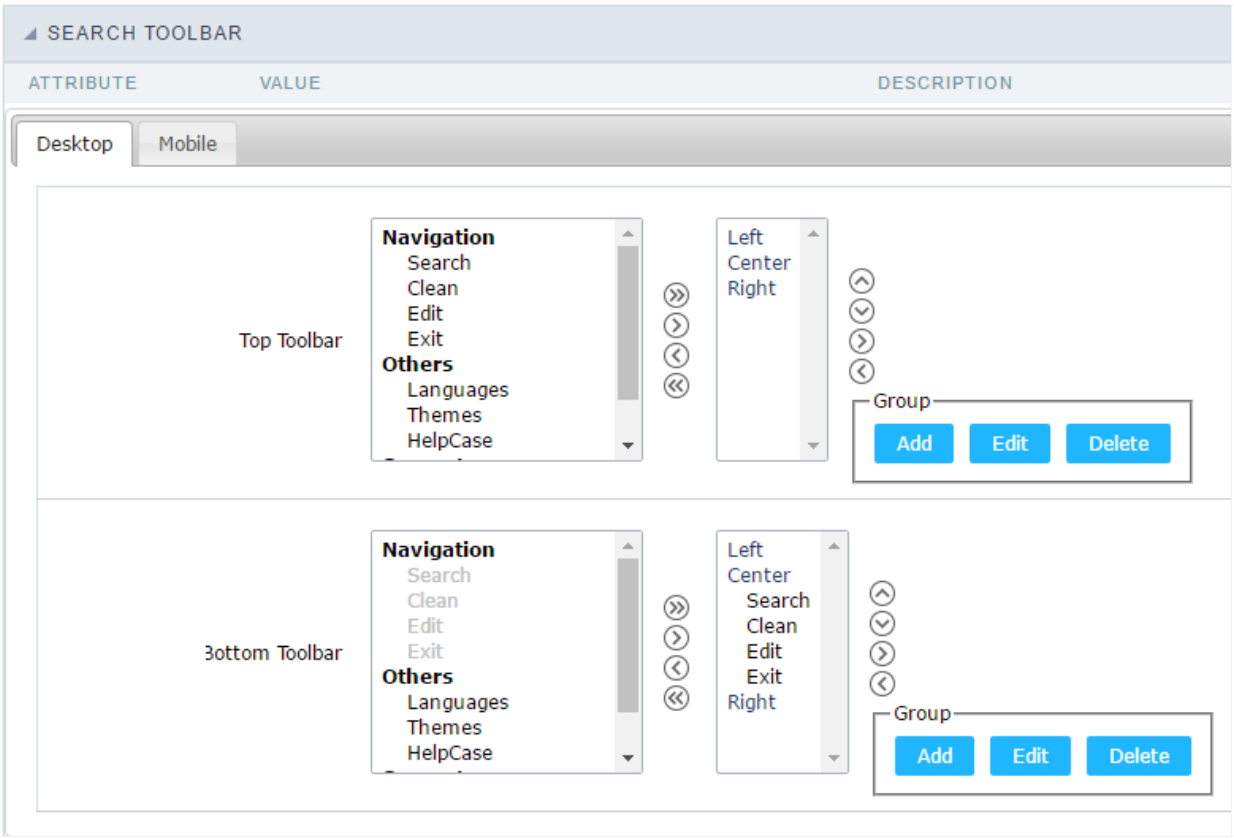
- **Marker position** : Marker’s position relative to the field.
- **Display message** : Displays whether or not the validation error message.

Toobar

Search Toolbar

The Search toolbar is divided in two parts: Top and Bottom, in a way that is possible to define the buttons that will be displayed in both bars. The selection of buttons in the top and bottom toolbar works independently, allowing the buttons to be displayed in

both bars at the same time.



Toolbar Interface.

Navigation:

Groups the options relative to the navigation buttons that can be displayed in the application.

- ***Search***: Execute the search.
- ***Clean***: Clean the all the search fields.
- ***Edit***: Enable the ***Save Tag*** option.
- ***Exit***: Exit the application.

Others:

Groups a diversity of options relative to the application.

- ***Languages***: Displays a combobox with the names available, defined in the project properties.
- ***Themes***: Displays a combobox with the themes available, defined in the project properties.
- ***HelpCase***: Displays a button to redirect to the help page.

Separator

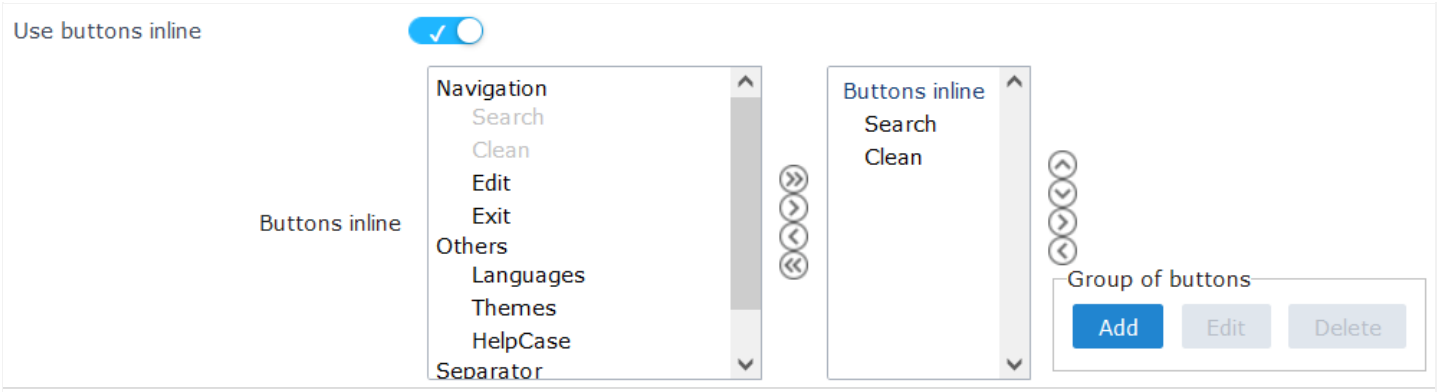
- **−−−−−−**: Dispays a line separating the buttons, when used the Group Buttons.

Use in-line buttons:

Allows the alignment of the filter buttons next to the fields.

Inline buttons: Allows you to select which buttons will be displayed next to the field, and you can sort them according to your wishes. This option is available by enabling **Use buttons inline** in the button settings.

When activating the **Inline Buttons** option, the screen to configure the buttons will be displayed.



The buttons available in this in-line button option are the same as those shown in the standard toolbar described above.

This is the result when using a radio button on the line

Button Settings

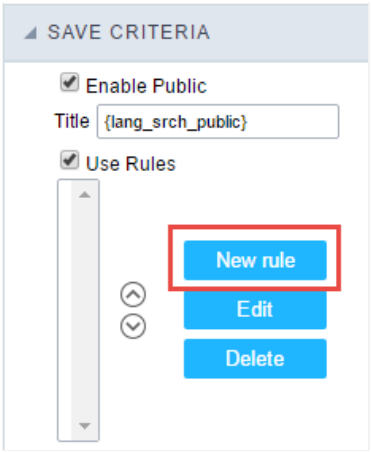
Button Settings Interface.

- **Hotkey**: Allows you to set keyboard hotkeys to a button.
- **Position of the in-line buttons**: Sets the positioning of the buttons to the right or left of the fields.
- **Column Quantity**: Sets the number of display columns of the buttons, allowing you to configure whether they will be displayed side-by-side or distributed in columns.

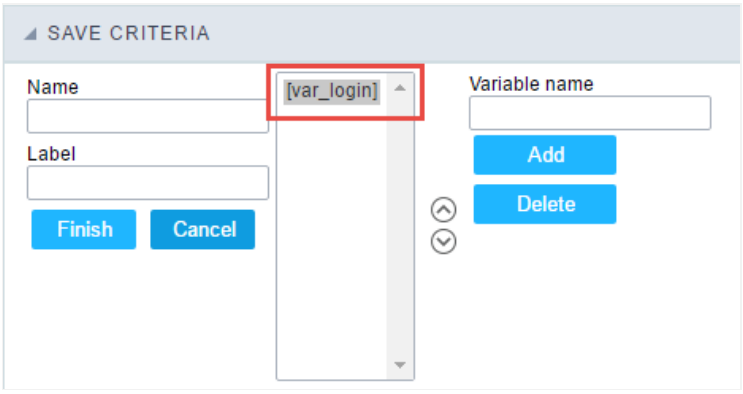
Options ![Options Interface.][barra_ferramentas_filtro_opcoes] *Options Interface.* * __Button Position(Top/Bottom)__ Positioning the buttons of the toolbar Top/Bottom. -->

Save Search

This feature allows the end-user to save his searches in a profile. You can create some rules, like to save the searches by user login.



Save Filter Interface.



Save Filter Interface.

Events

In event blocks can be used global variables, local, JavaScript code, CSS codes and Scriptcase macros.

onScriptInit

This event occurs when the application its loaded, or reload. Occurs before the application runs the SQL statement, so using this event it is possible to modify the grid SQL statement dynamically, based on any logical. Macros frequently called on this event: sc_select_field, sc_select_order, sc_select_where(add), etc.

onRefresh

This event occurs when the form application is reloaded, is possible to reload the form based on fields (select , radio, checkbox)

onSave

This event occurs always that some search setting is saved in the filter application.

onValidate

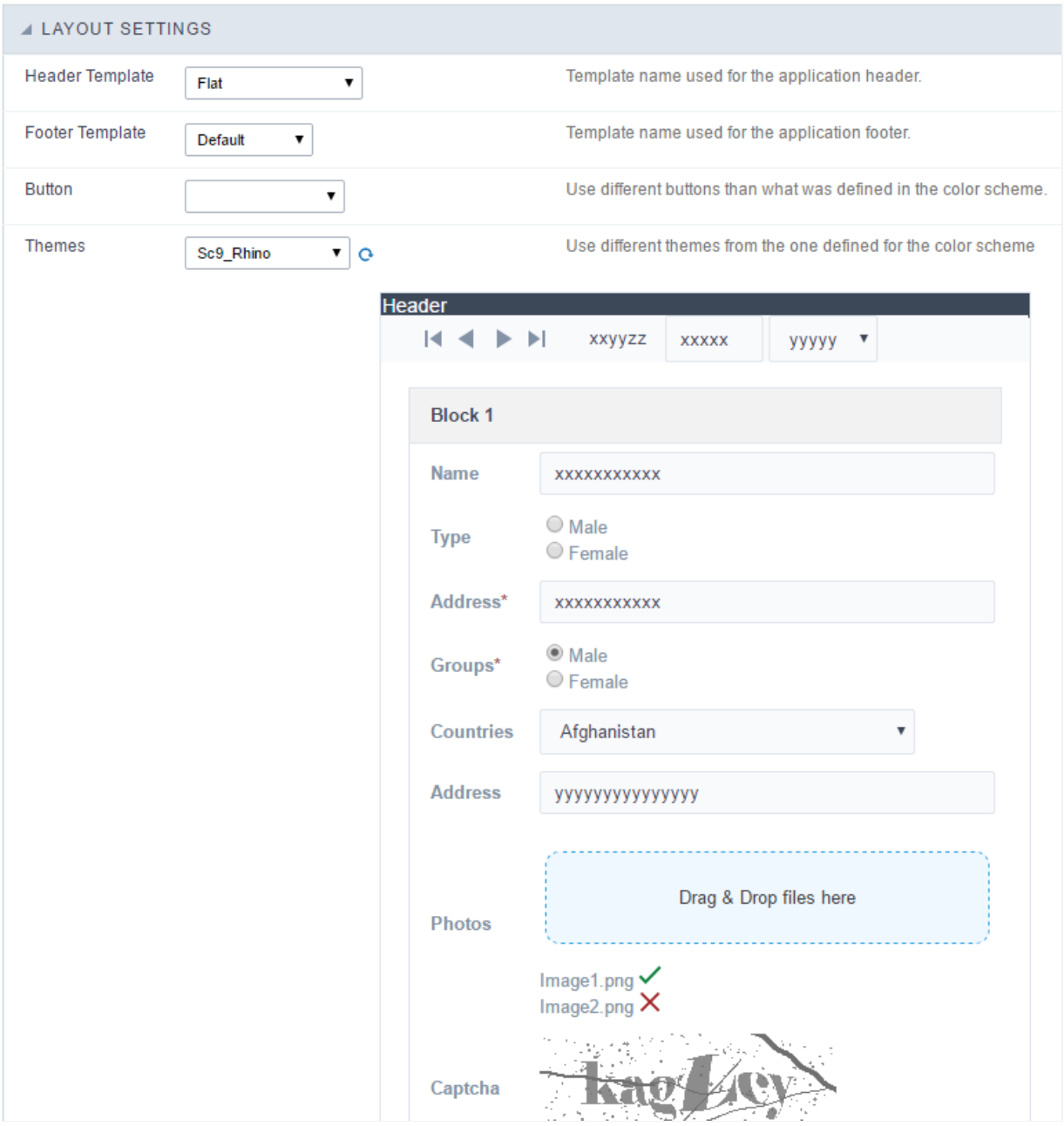
This event occours when the search form is submitted. Ex. in order to prevent some searches or to validate any search params if(strlen({name}) <= 0) { sc_error_message(" Please fill the name field before submit your search "); }

Layout

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout Settings

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

Allows choosing the button theme for the application.

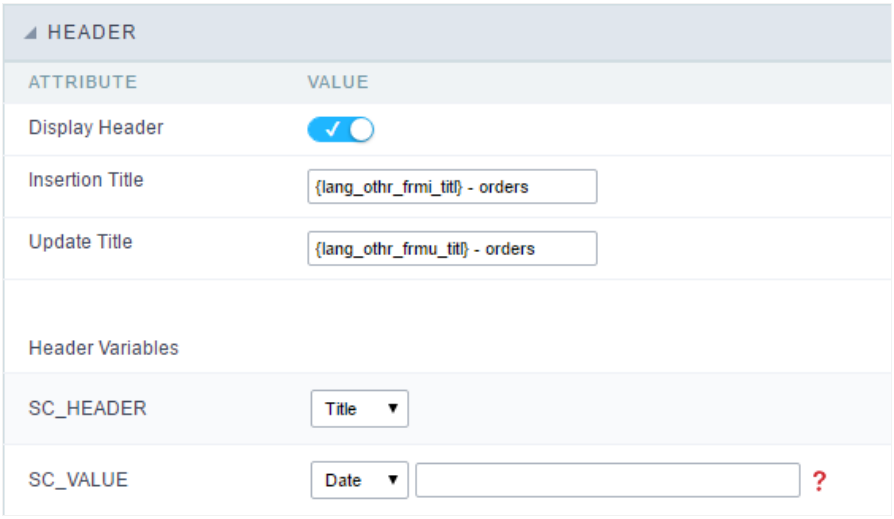
Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header & Footer

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

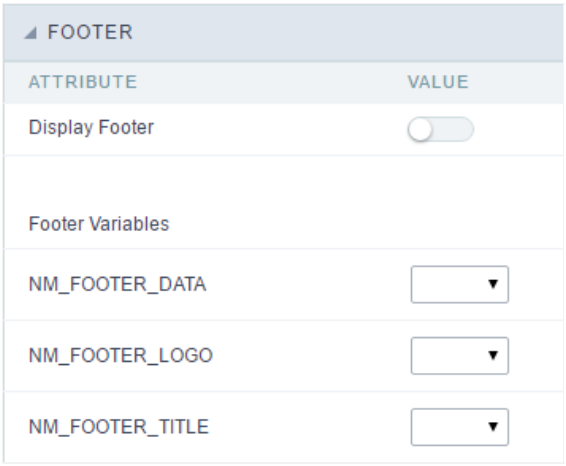
Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header.
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Fields

This is a list of fields available for use in the filter. Click on the desired field to access the configuration instructions.

Text Field	Field Select
Integer Field	Field Double Select
Decimal Field	Field Checkbox
Currency Field	Field Radio
Date Field	Field Text Auto-Complete
Field time	Field Number Auto-Complete
Field Datetime	

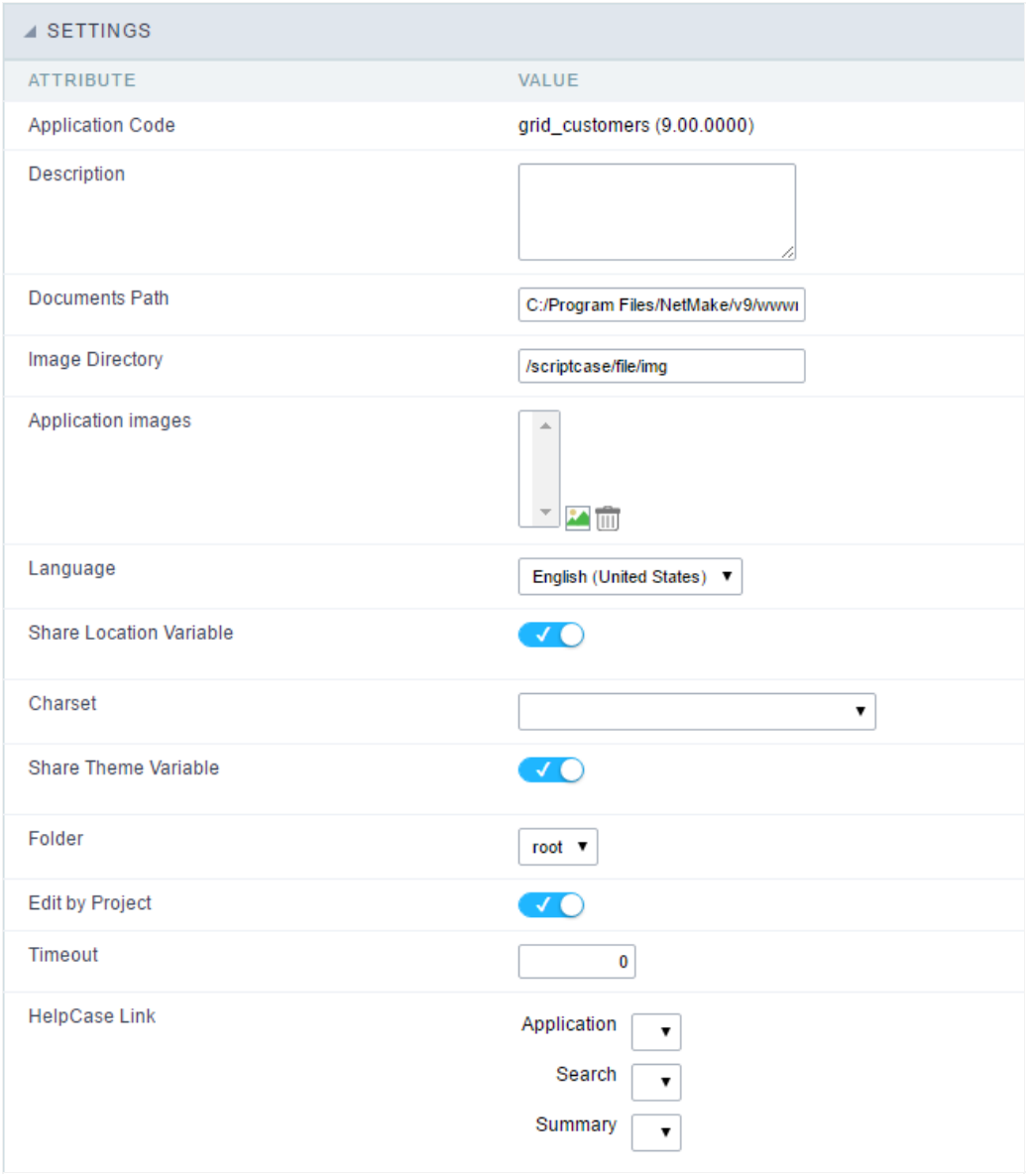
Related Links

Related Videos ▷

Application Settings (PDF Report)

Settings

With this interface, you can set the common attributes of the app.

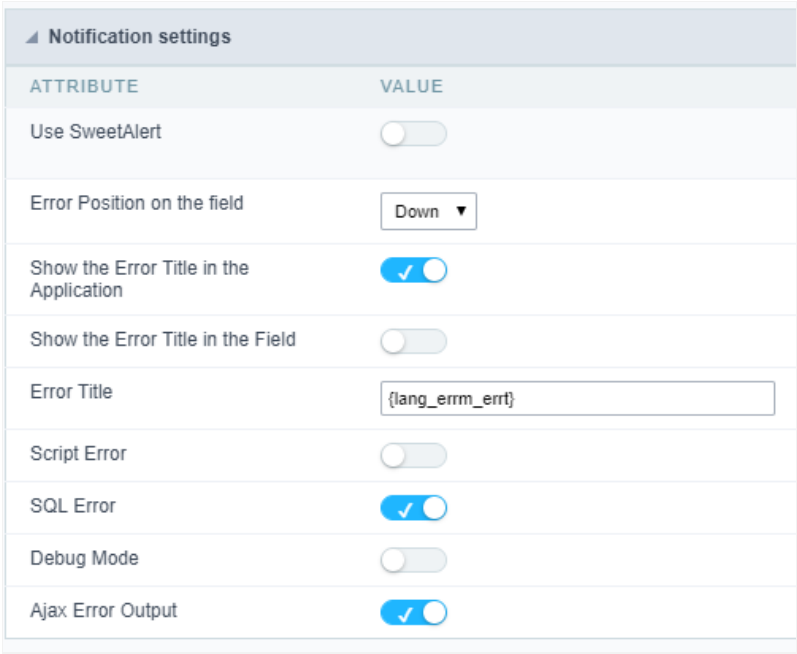


Application Settings Interface

- **Attributes**
  - **Application Code** : It is the name that defines an application. An app can be renamed at the List of Application.
  - **Description** : This field contains a brief description of the application objectives.
  - **Documents Path** : The absolute path to store uploaded documents in the application.
  - **Image Directory** : The filesystem directory to store the application images.
  - **Application images** : Import images into the application to allows using them in the application.
  - **Language** : Set the default language of the application. Display all the application hints and messages in the selected language.
  - **Share Location Variable** : Define if the app shares the regional settings with other applications through a session variable.
  - **Charset** : Define a specific charset to use in the application.
  - **Share Theme Variable** : Define if the app shares the Theme settings with other applications through a session variable.
  - **Folder** : Define the project folder that contains the app.

- **Edit by Project** : Define if other project developers can edit the application.
- **Timeout** : Set the session runtime timeout in seconds. If the value is Zero, it assumes the default timeout of the PHP.
- **HelpCase Link** : It allows to associate a HelpCase file with the application.

Notification Settings



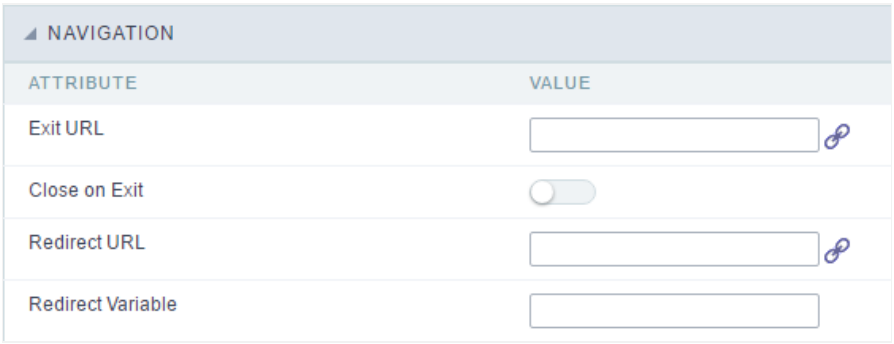
- **Use SweetAlert**: Use the SweetAlert to display messages from the application. When this option is active, it will replace the browser's "confirm" and "alert".
- **SweetAlert position using Toast** : The position to display error messages on the application.

- **Error Position on the field** : The position to display error messages when criticizing the field.
- **Show the Error Title in the Application** : Define to display the title line of the error message or not.
- **Show the Error Title in the Field** : Define to display the title line of the error message in the field or not.

- **Script Error** : Allows displaying the line code where there is an error..
- **SQL Error** : Allows displaying the SQL statement if it got an error.
- **Debug Mode** : Runs the application in Debug mode, showing all SQL statements the application is executing.
- **Ajax Error Output** : Enables the Ajax alert for debugging errors.

Navigation

This interface allows defining the navigating behavior of the application



Navigation Interface.

Exit URL

URL to where the user goes when he clicks on the “exit” button.

Close on Exit

Close the browser window when the user clicks on the “exit” button.

Redirect URL

Redirect to another URL in case there aren’t any global variables available.

Redirect Variable

Creates a variable with the application name and sends it to the redirected application.

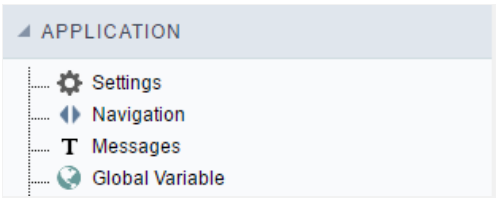
Global Variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**“**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target=”_blank”} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



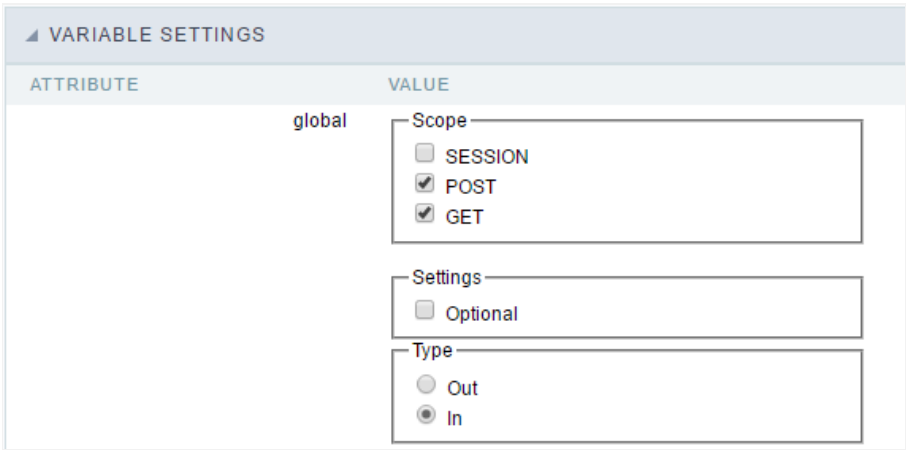
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

Indicates where the application is using the variable.

Related Links 🔗

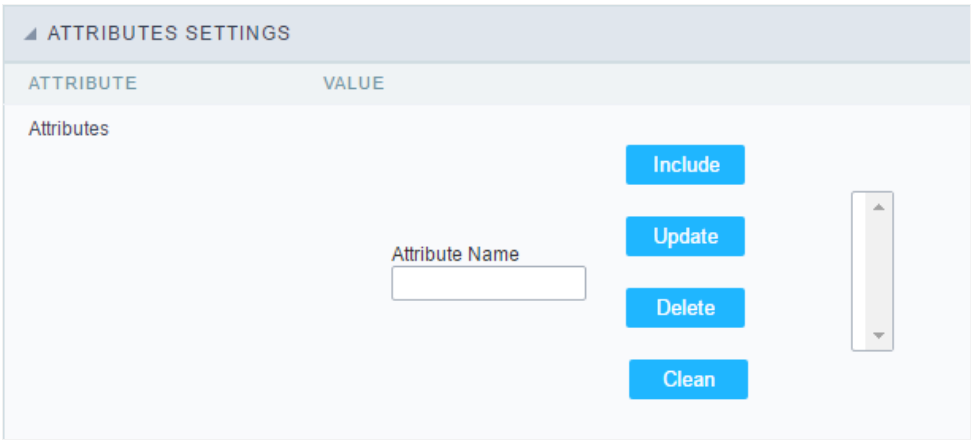
Related Videos ▷

PDF Report Programming

In this version of ScriptCase is incorporated the concept of programming with the use of attributes, methods, resources and libraries. In the previous version it was already possible to create business rules in applications using this concept. The big difference now is that this can be done in a more organized, facilitating both the development as the understanding of the rule by another developer.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal Libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

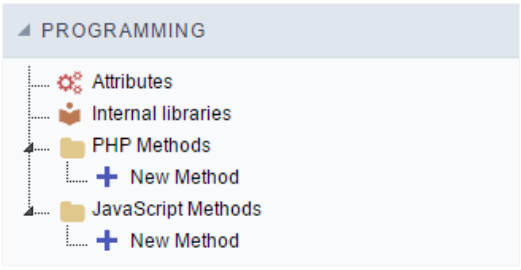
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

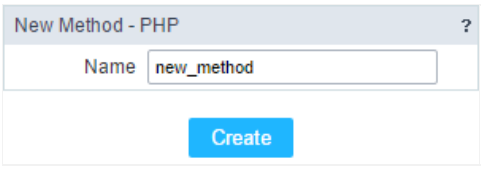
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

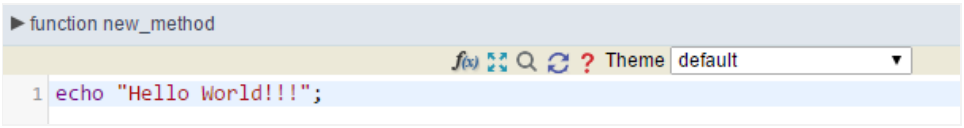


- Give the method a name and click Create. As in the image below.

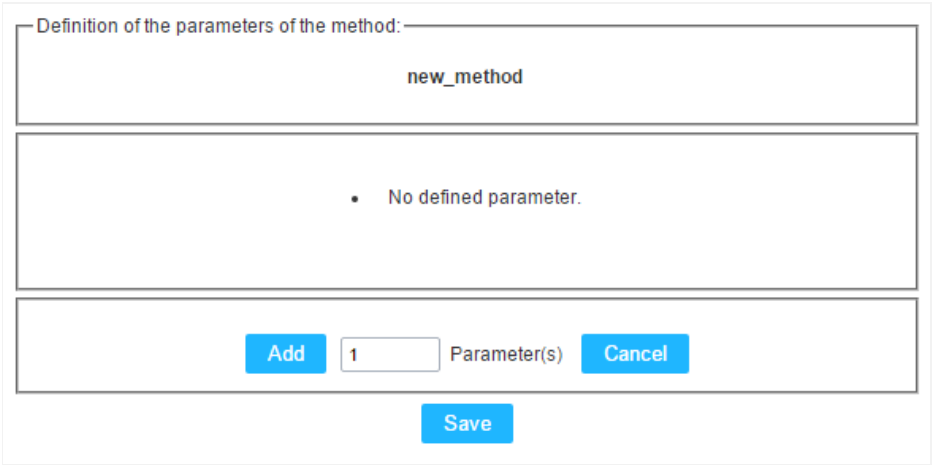
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



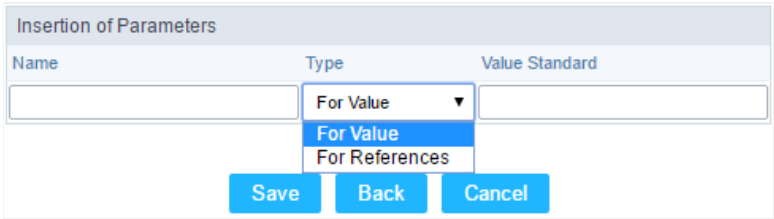
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

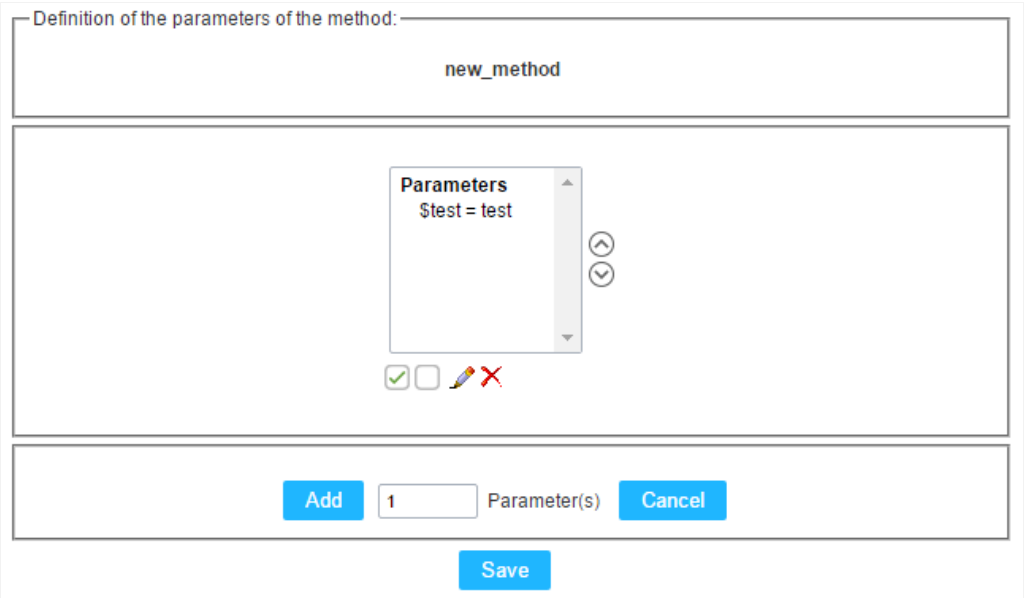


- **Name** : Type in the variable's name.

- **Type** : Selecting the type of variables: For Value or For Reference.
- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:

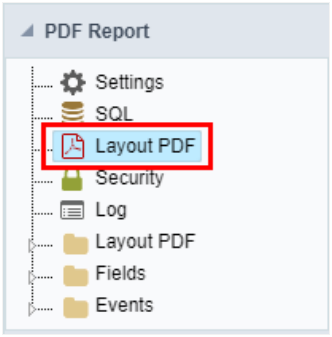


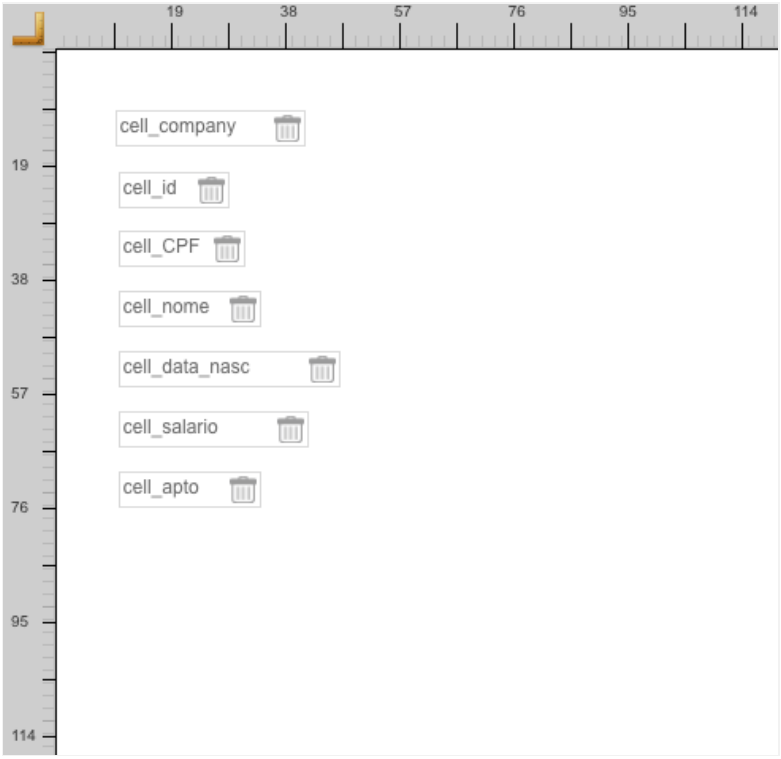
- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

PDF Report Layout PDF

The PDF Report application has a way of positioning the report fields in a simple and dynamic way, using the drag and drop functionality.

In the PDF Layout, at the main screen, you can change some application settings. These settings are also present in PDF Report > Settings.

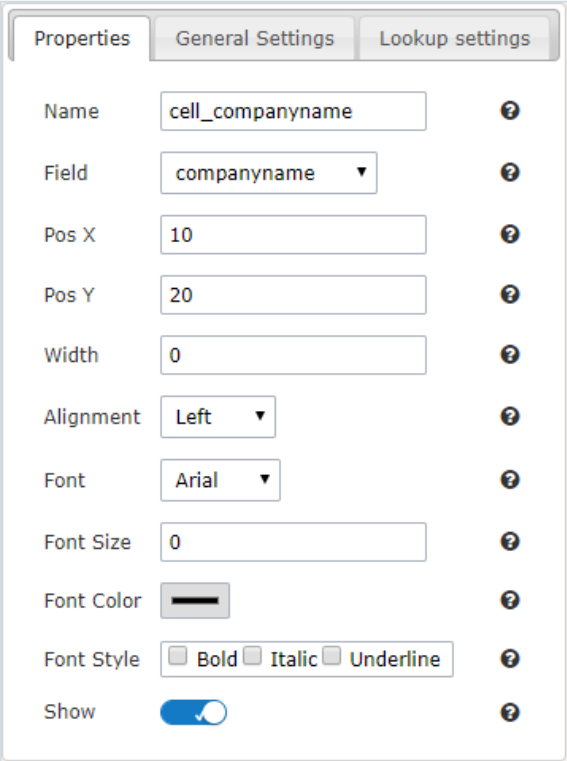




Using Drag and Drop functionality, you can configure the placement of fields in this interface. To change them you need to click on the field you prefer to change the position and drag to their new position.

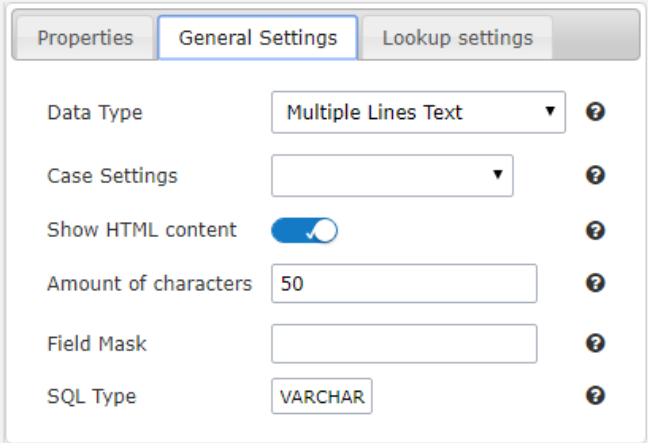
Clicking on a specific field opens a field configuration screen.

Properties



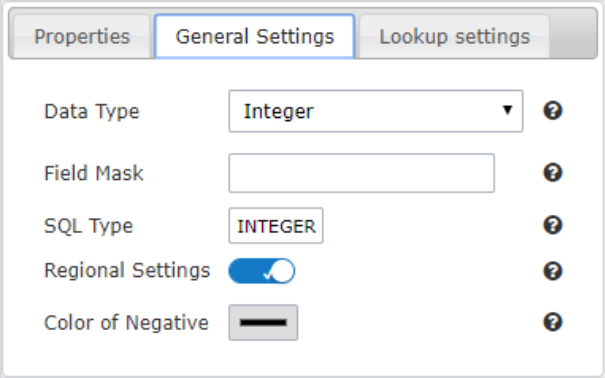
- **Name**: On this field, it is necessary to inform a name to identify the field.
- **Field**: On this option, you need to select the field from the table that will be used in the cell. In case you have selected a field from the "Special" category you will see the following:
  - **System Date**: The current date of the system on which the application runs.
  - **Current Page**: Number of the current page that the cell is located.
  - **Total Page**: Total number of application pages.
  - **Label**: Selecting this option will open an input so you can inform the text that will be displayed.
- **Pos X**: Positioning the X axis. Up and down.
- **Pos Y**: Positioning the Y axis. Left and Right.
- **Width**: Enter the width of the cell. The unit of value is pixel.
- **Alignment**: In this field, you will select the alignment of the cell text: Left, Right or Center.
- **Font**: In this field, you will select a source according to the list.
- **Font Size**: Here you inform the font size of the text that will be displayed.
- **Font Color**: In this field, you will inform a color in hexadecimal. (Example: # 000000)
- **Font Style**: In this field, you will mark the style of the font: Bold, Italic and / or Underline. (You can select more than one option)
- **Show**: In this field, you enable or disable the field display.

General Setting - Text Fields



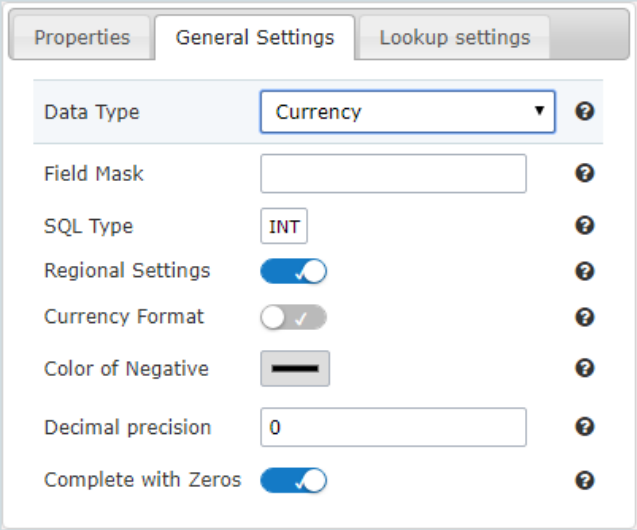
- **Data Type** : Define the type of field for the application.
- **Case Settings** : Convert the letter from the field when losing focus. The options are:
  - **Upper case** : All in Upper Case.
  - **Lower case** : All in Lower case.
  - **Capitalize first word** : Capitalizes the first letter of the first word.
  - **Capitalize all words** : Capitalizes the first letter of all the the words.
- **Show HTML content** : Determines if the HTML contained in the field will be displayed or not. If enabled, the html will be displayed, otherwise the html will be interpreted by the browser.
- **Amount of characters**: Defines the field width. This option is available only for the field type **Multiple lines text**.
- **Field Mask**: In this field, you can configure a display mask according to the table given in Applications> Grid > Fields> Text.
- **SQL Type** : Informs the data type of field in the database.

General Setting - Integer Fields



- **Data Type** : Define the type of field for the application.
- **Field Mask**: In this field, you can configure a display mask according to the table given in Applications> Grid > Fields> Text.
- **SQL Type** : Informs the data type of field in the database.
- **Regional Settings** : Allows to apply the Regional Settings to format of the fields. When not enabled, you will view attributes for set yourself. You can change of the Regional Settings in **Locales >Regional settings**. For more information click here.
- **Color of Negative**: In this field, you will inform a color in hexadecimal. (Example: # 000000)

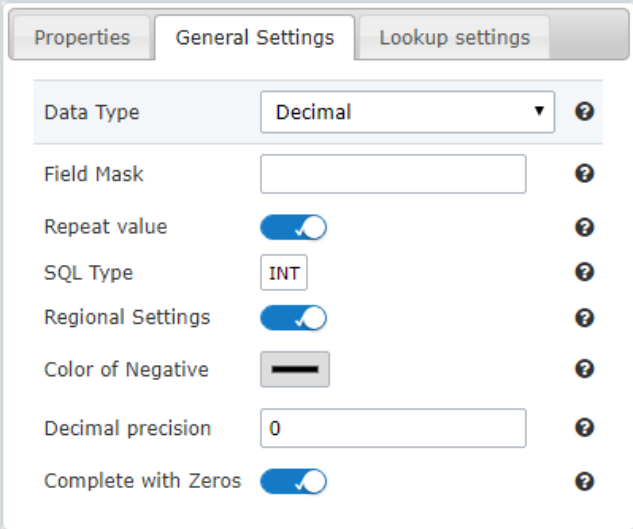
General Setting - Currency Fields



- **Data Type** : Define the type of field for the application.
- **Field Mask**: In this field, you can configure a display mask according to the table given in Applications> Grid > Fields> Text.

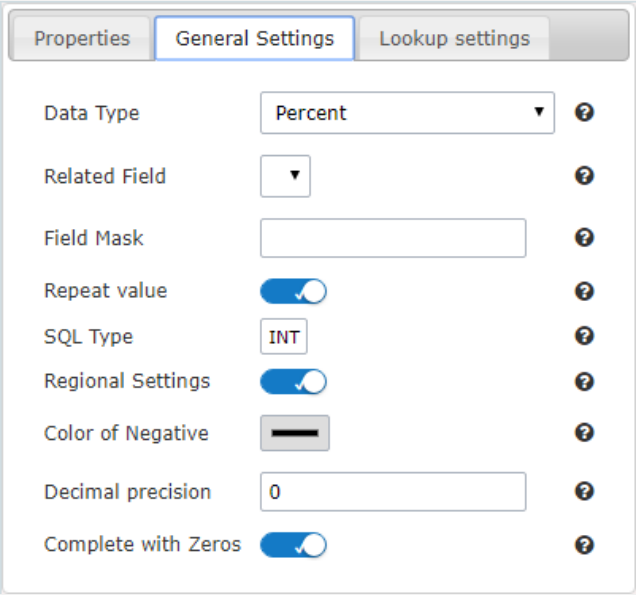
- **SQL Type** : Informs the data type of field in the database.
- **Regional Settings** : Allows to apply the Regional Settings to format of the fields. When not enabled, you will view attributes for set yourself. You can change of the Regional Settings in **Locales >Regional settings**. For more information click here.
- **Currency Format** : Defines the content of the field that if it will be presented with the currency format.
- **Color of Negative**: In this field, you will inform a color in hexadecimal. (Example: # 000000)
- **Decimal Precision** : Number of decimal places. This value is part of the field size.
- **Complete with Zeros** : Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

General Setting - Decimal Fields



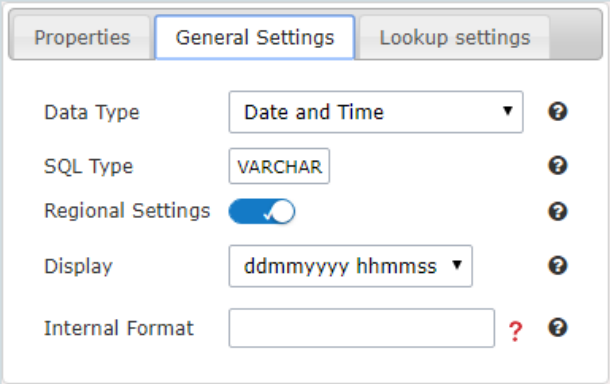
- **Data Type** : Define the type of field for the application.
- **Field Mask**: In this field, you can configure a display mask according to the table given in Applications> Grid > Fields> Text.
- **SQL Type** : Informs the data type of field in the database.
- **Regional Settings** : Allows to apply the Regional Settings to format of the fields. When not enabled, you will view attributes for set yourself. You can change of the Regional Settings in **Locales >Regional settings**. For more information click here.
- **Color of Negative**: In this field, you will inform a color in hexadecimal. (Example: # 000000)
- **Decimal Precision** : Number of decimal places. This value is part of the field size.
- **Complete with Zeros** : Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

General Setting - Percent Fields



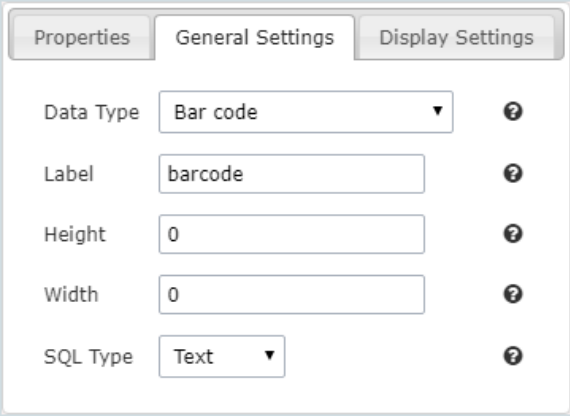
- **Data Type** : Define the type of field for the application.
- **Related Field**: Field that contains the values for the calculation of the percentage.
- **Field Mask**: In this field, you can configure a display mask according to the table given in Applications> Grid > Fields> Text.
- **SQL Type** : Informs the data type of field in the database.
- **Color of Negative**: In this field, you will inform a color in hexadecimal. (Example: # 000000)

General Settings - Date/Time Fields



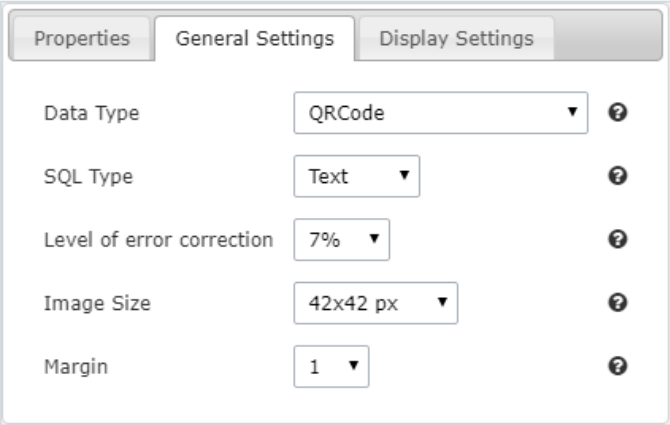
- **Data Type** : Define the type of field for the application.
- **SQL Type** : Informs the data type of field in the database.
- **Regional Settings** : Allows to apply the Regional Settings to format of the fields. When not enabled, you will view attributes for set yourself. You can change of the Regional Settings in **Locales >Regional settings**. For more information click here.
- **Display**: Offers a series of formats predefined for displaying dates.
- **Internal Format**: Defines the internal format of the fields (for database storage) date, time, and datetime if the field type in SQL is varchar type.

General Settings - Bar code Fields



- **Data Type** : Define the type of field for the application.
- **Height**: Sets the height in pixel of the image display. Must contain only numbers.
- **Width**: Sets the width in pixel of the image display. Must contain only numbers.
- **SQL Type** : Informs the data type of field in the database.

General Settings - QRCode Fields



- **Data Type** : Define the type of field for the application.
- **SQL Type** : Informs the data type of field in the database.
- **Level of error correction**: Codewords are 8 bits long and use the Reed–Solomon error correction algorithm with four error correction levels. The higher the error correction level, the less storage capacity.
- **Image Size**: Lets you set the display size of the QR Code by selecting one of the available sizes.
- **Margin**: Lets you set the margin width around the QR Code by selecting one of the available sizes.

Related Links 🔗

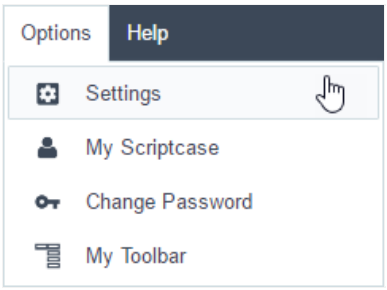
Related Videos ▷

Creating a PDF Report Application

New Application

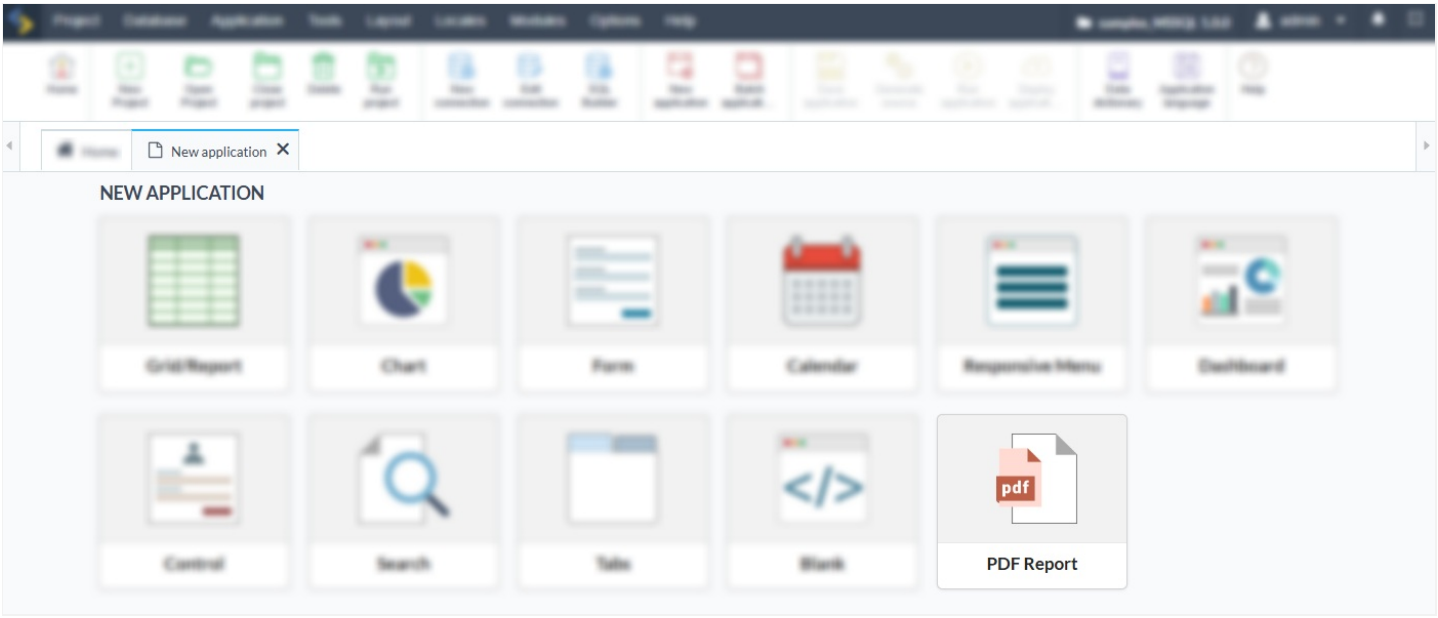
With the growing evolution of the Grid/Report application's features for generating reports, the old PDF Report application was removed from the interface. However, it is still possible to use it.

To enable this functionality, the Scriptcase administrator must access the **Options > Settings** menu and open the System Settings page.

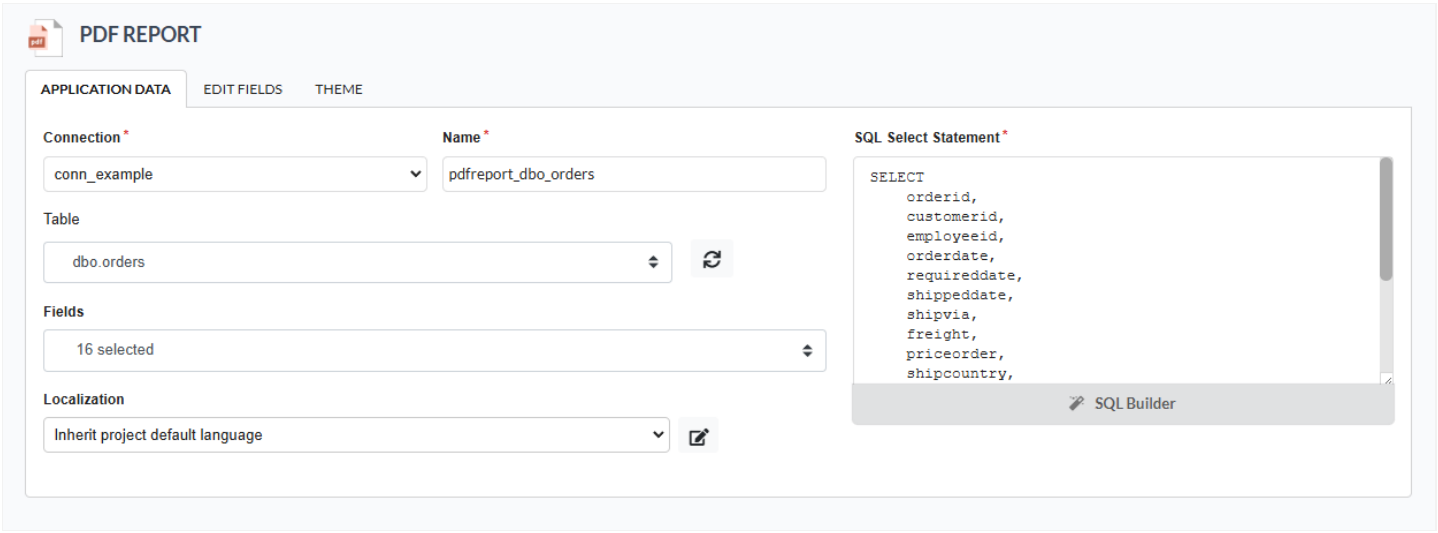


At the bottom of the page, after the foreign key configuration section, you can enable the old menu application to display it as an option when creating new applications.

Once enabled, the application will appear on the new application creation screen, as shown in the image below.



Application Data



Connection

Defines the connection used for the creation of the application. The existing connections in the project will be listed.

Name

The name of the application being created. It cannot contain special characters.

Table

Defines the tables to be used in the application. (Form and Calendar can only use one table).

Fields

Defines the fields that will be part of the applications.

Localization

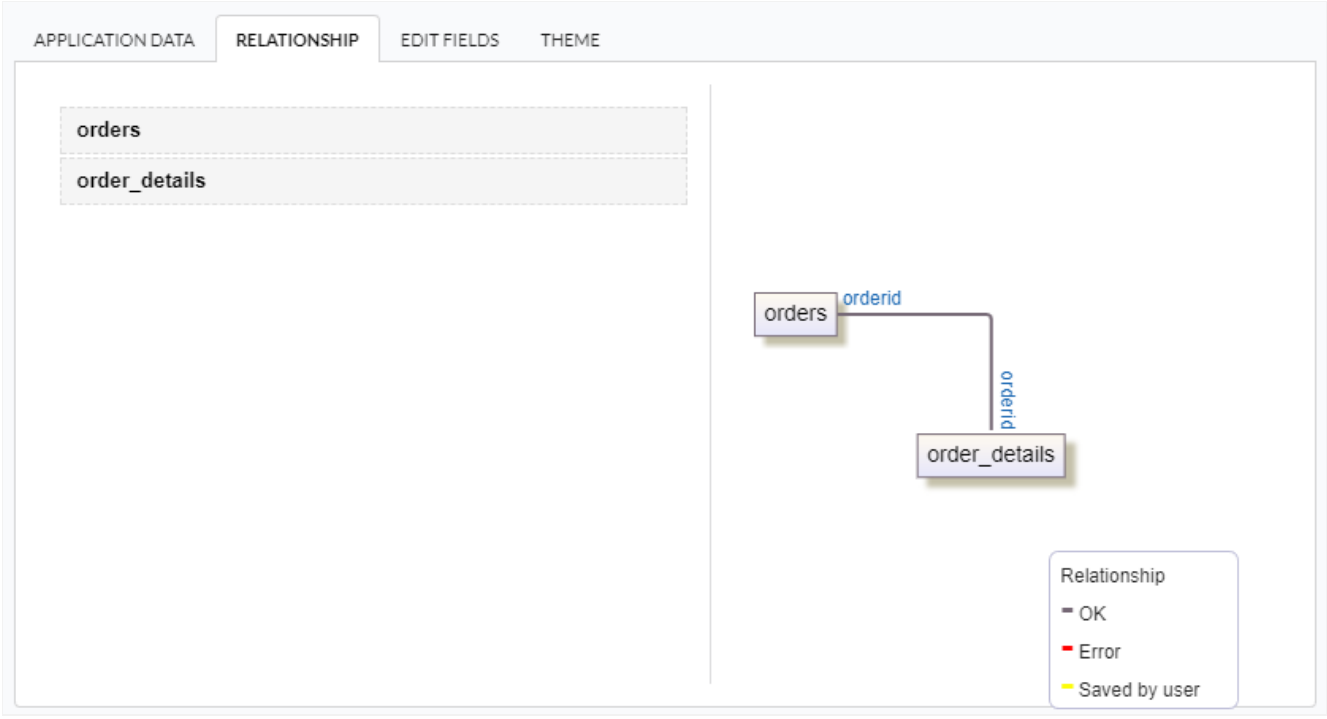
The language of the application to be created. The project's default language is automatically selected.

SQL

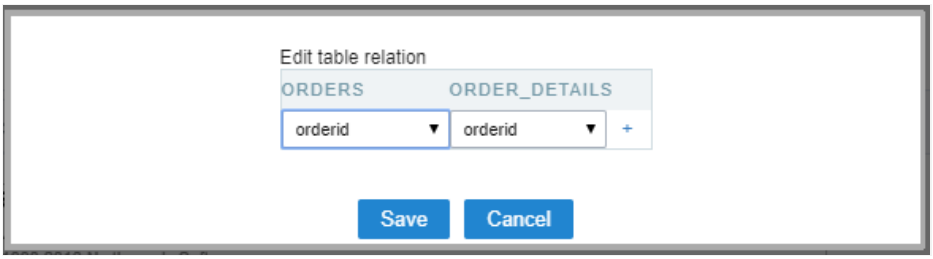
Displays the SELECT command generated after selecting the tables and fields. This field also allows the insertion of a previously created SQL statement, as long as it uses tables that exist in the database selected in the connection.

Relationship

When select two or more tables, the tab “Relationship” will be displayed. In this tab we can see the relationship created between the tables, where we can edit the related fields.

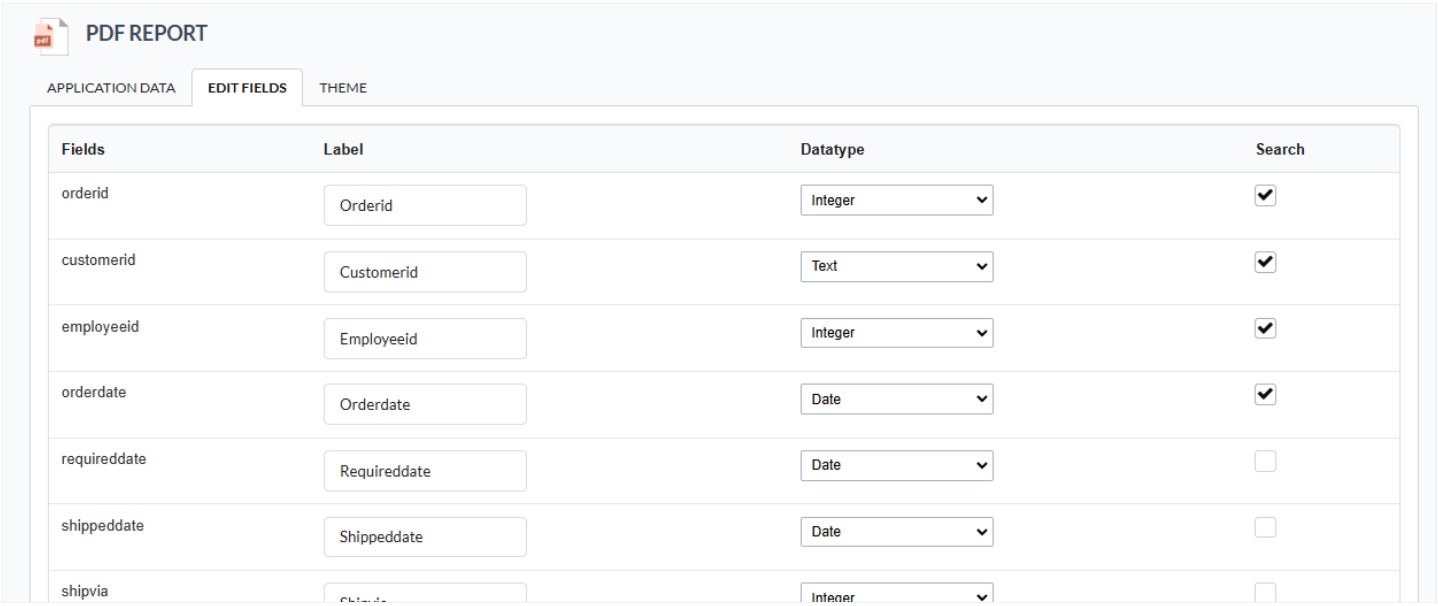


When we click in a link, in the screen above, it will displayed the related field’s edition form, as you can see in the image below.



Edit Fields

This screen displays the fields of the selected tables and allows adjustments to be made before creating the application, such as changing the data type, display name, and other configurations.



Fields

Names of the database fields.

Label

Names of the fields in the generated application's interface.

Datatype

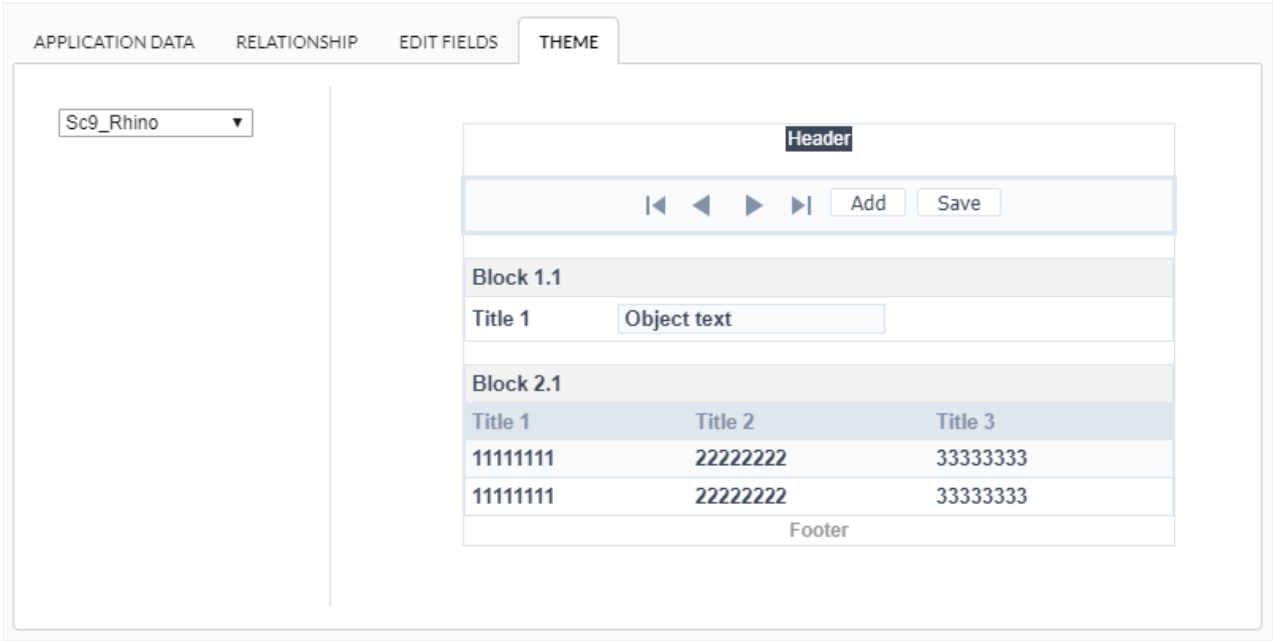
Specifies the field's data type.

Search

Defines the fields available in the filter.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.

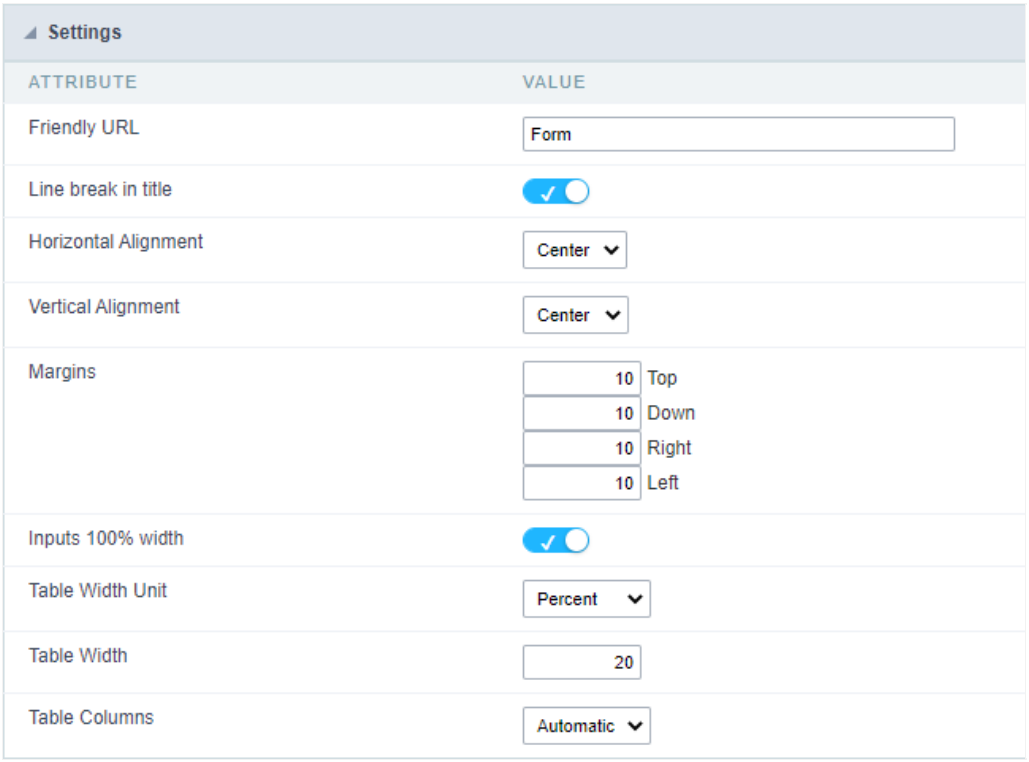


Related Links 🔗

Related Videos ▷

Form Calendar Settings

This interface is useful for editing the main display settings of the app.



Form Settings Interface.

Friendly URL

Allows to define a URL for the application different from the application name. You can use the following chars (a-z, A-Z, 0-9, -_). You still can change it on the initial screen of the home project, through the “Friendly URL” column in the apps list.

Line break in title

Use it if you want to break the line on the field titles.

Horizontal Alignment

Allows you to set the alignment of the application on the page.

Vertical Alignment

Allows you to define the initial vertical alignment of the application (Above, Centered and Below).

Margins

Allows to define the margins of the application (Right, Left, Up and Down) in pixels.

Table Width

The width of the form table. Scriptcase uses Plain HTML to generate applications by using tables lines and cells.

Table Width Unit

Measurement unit for the table width defined in the previous option, being: percentage, pixel, or automatic.

Table Columns

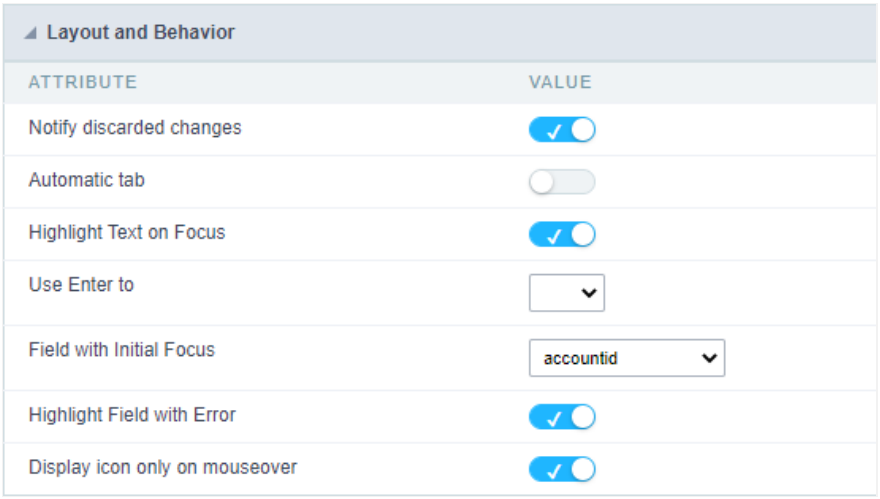
This parameter defines the column (fields) width of the table (application).

Labels width

When the previous option is set to “Provided” you must inform the width of the labels here.

Layout and Behavior

This interface allows setting the behavior of the app.



Layout and Behavior configuration Interface.

Notify discarded changes

Notifies the user when any changes made will be lost when reloading the data.

Automatic tab

Changes the focus to the next field when the amount of characters reaches its defined limit.

Highlight Text on Focus

Highlights the field when selected.

Use Enter to

Allows to use the “Enter” key to pass the focus to the next field.

Field with Initial Focus

Determines the field starts with focus when accessing the application. This option doesn’t work with fields that contain a watermark.

Highlight Field with Error

Focus the field with the error when submitting the form.

Use a template from the HTML Editor

Allows to use the TinyMCE editor. You can edit and create your HTML Templates.

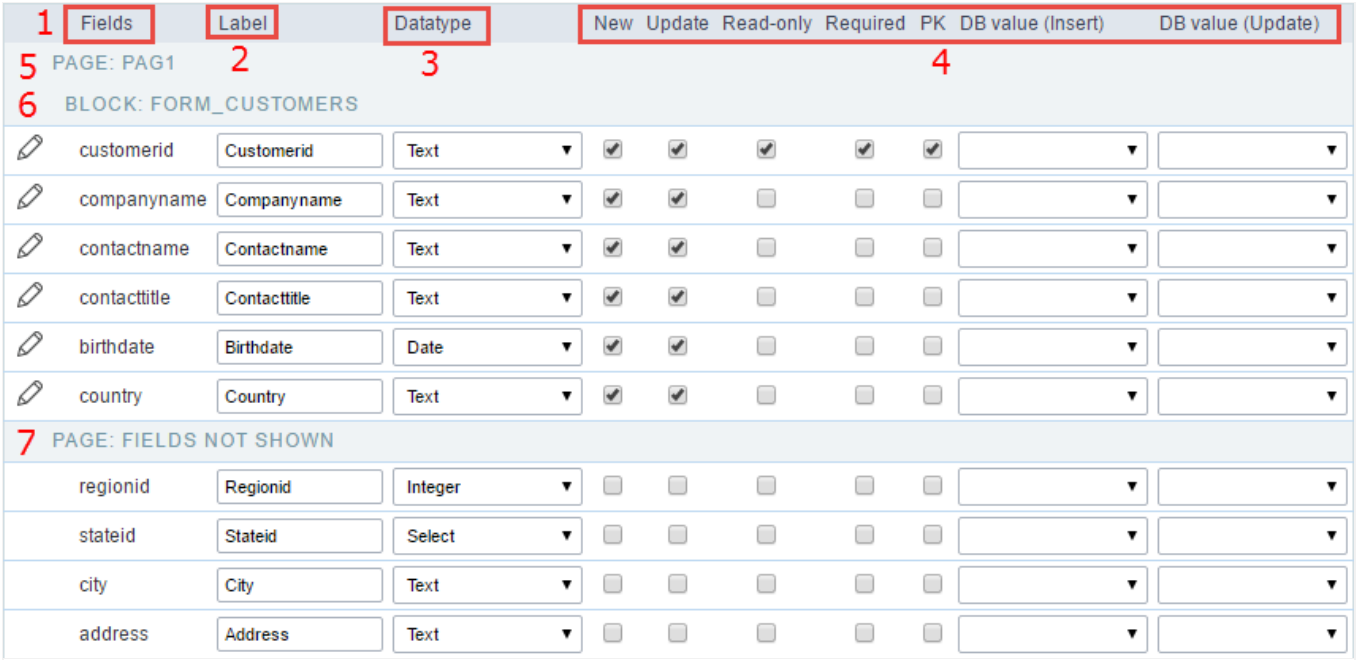
Related Links

Related Videos ▷

Calendar Edit fields

This interface is useful for editing the field settings and their position to display.

Edit Fields



Edit Fields Configuration.

Fields

It allows accessing the field settings (a pencil icon on the left). You can change the field position by dragging them to the desired position. Drag a field to “fields not displayed” if you don’t want it in the app.

Label

It defines the title of the field in the app. For example: if the field name in the database is fld_txt_customer_name, you can display the label “Customer Name”.

Data type

It informs the data type of the field.

New

It defines if the field is available when inserting new records.

Update

It defines if the field is available when updating records.

Read-Only

It defines the field as a label. The user can’t change its value.

Required

It defines if the field must contain a value.

PK

It defines the Primary Keys fields.

DB value (Insert)

Defines a default value for the field when inserting a new record, like an auto-increment, Date, DateTime, or IP.

DB value (Update)

Defines a default value for the field when updating a record, like an auto-increment, Date, DateTime, or IP.

Page

It shows the pages available in the application. All apps have a page, at least. Each page contains one or more blocks.

Blocks

It shows the blocks available in the application. Blocks contain fields. All apps have a block, at least. A block is displayed if it contains one or more fields.

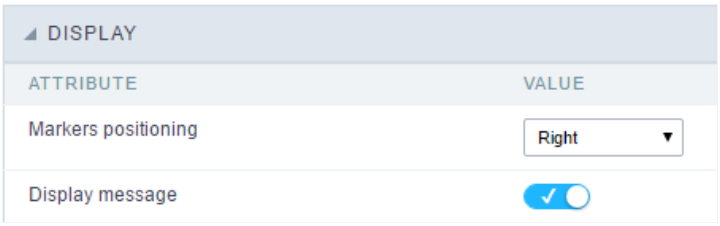
Page Fields Not Shown

Here we can see the fields that are not in the application.

Observe that you can drag any line to the desired position, blocks, and page. Pages contain Blocks, and Blocks contain fields.

Display Settings

Settings of messages display.



Display Interface.

Attributes #### Markers positioning Set the position of the markers that Indicates required fields.

Display message
Set it if you want to display the message of the required field.

Related Links 🔗

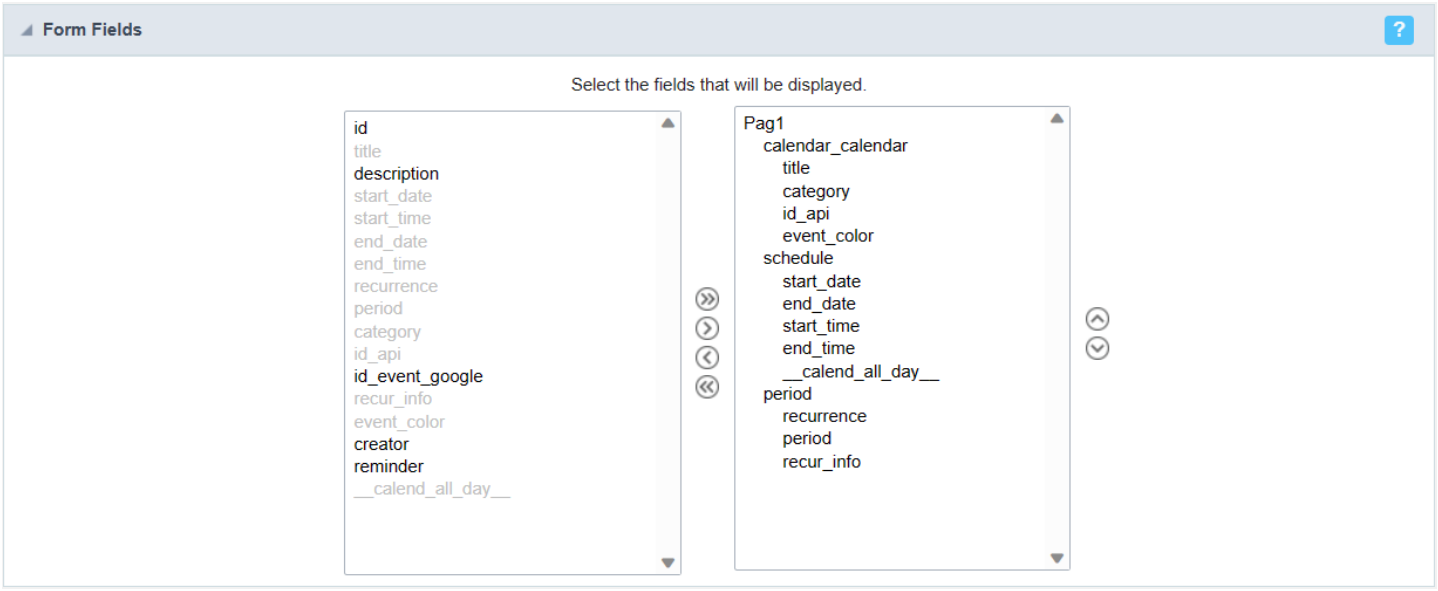
Related Videos ▷

Calendar Fields positioning

In this configuration screen we have a list of all the fields available in the application, whether they are fields mapped from the table or virtual fields (Created only in the Scriptcase interface).

It is also possible to allow the end user to manipulate the application's fields in the way they prefer, for that we must add the **columns button** in the application's toolbar.

In the **left column**, we have a list with all the fields available for use and in the **right column** we have a list of the fields that were selected to be displayed in the running application.



Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

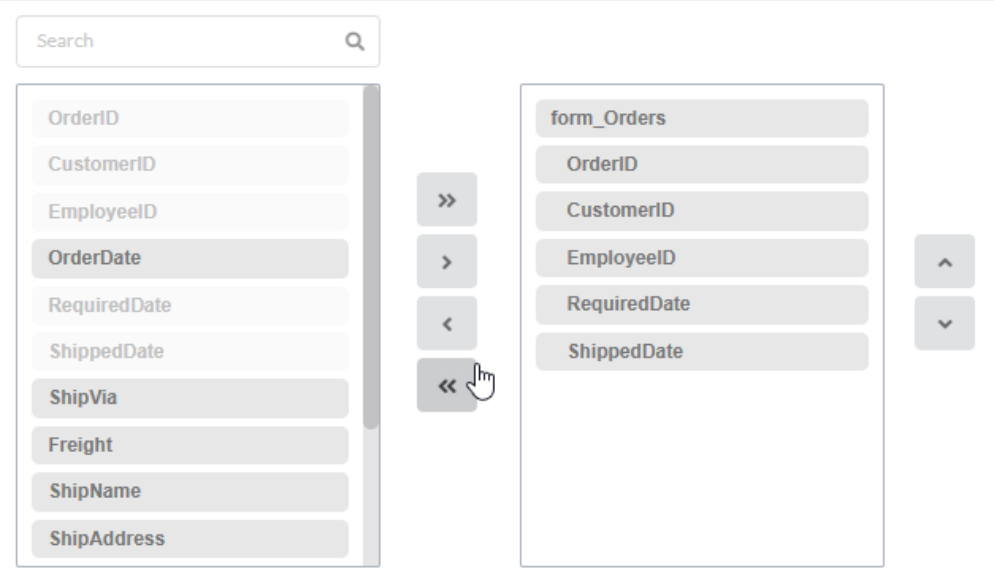
- » - Move all fields to the right.
- › - Moves only selected fields to the right
- ‹ - Moves only the selected fields to the left.
- « - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the › button.

To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning



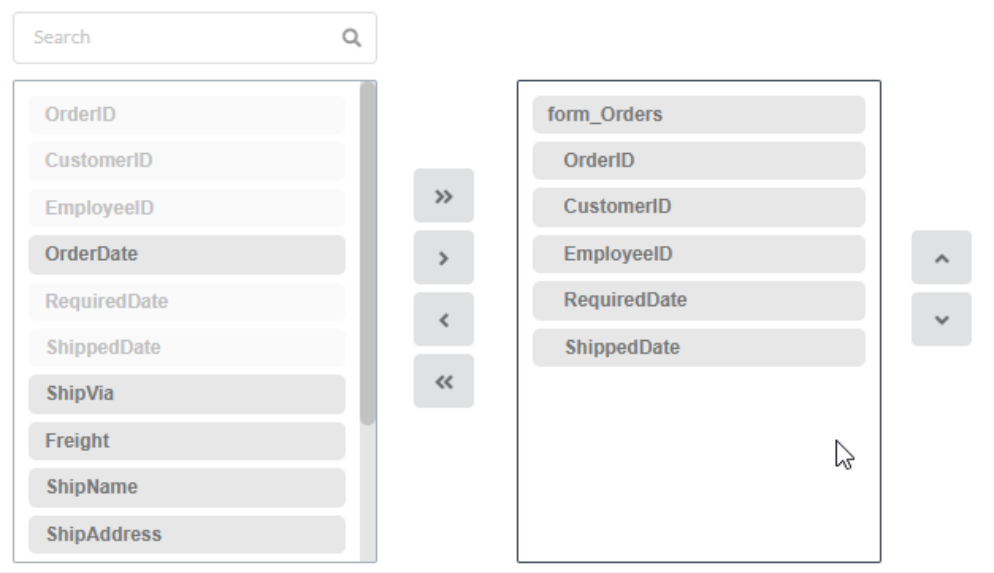
Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

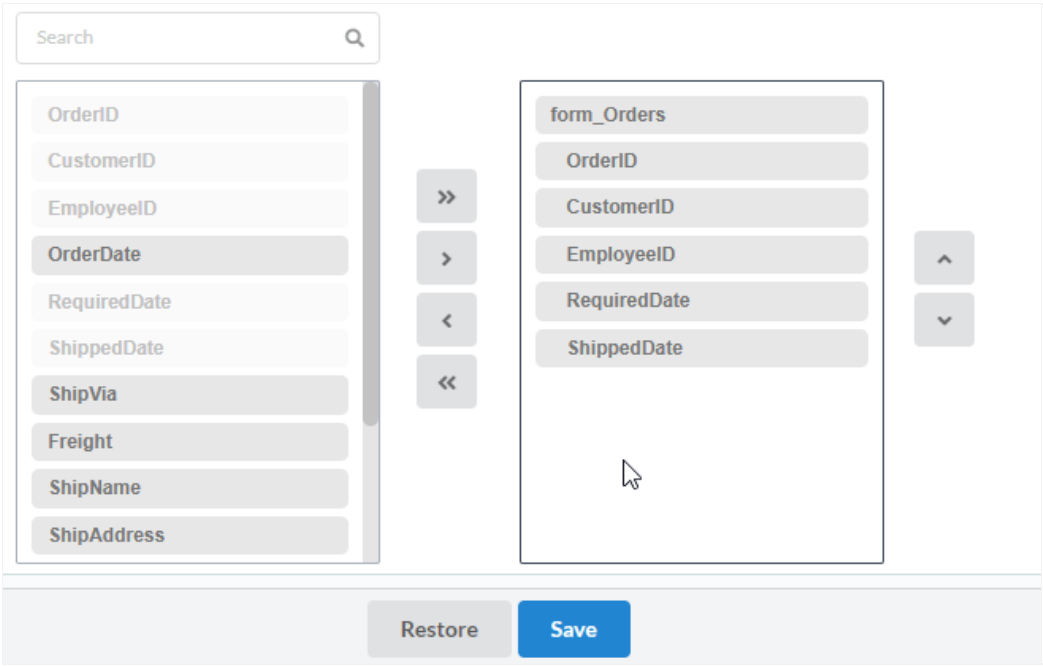


Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.



Related Links

Related Videos

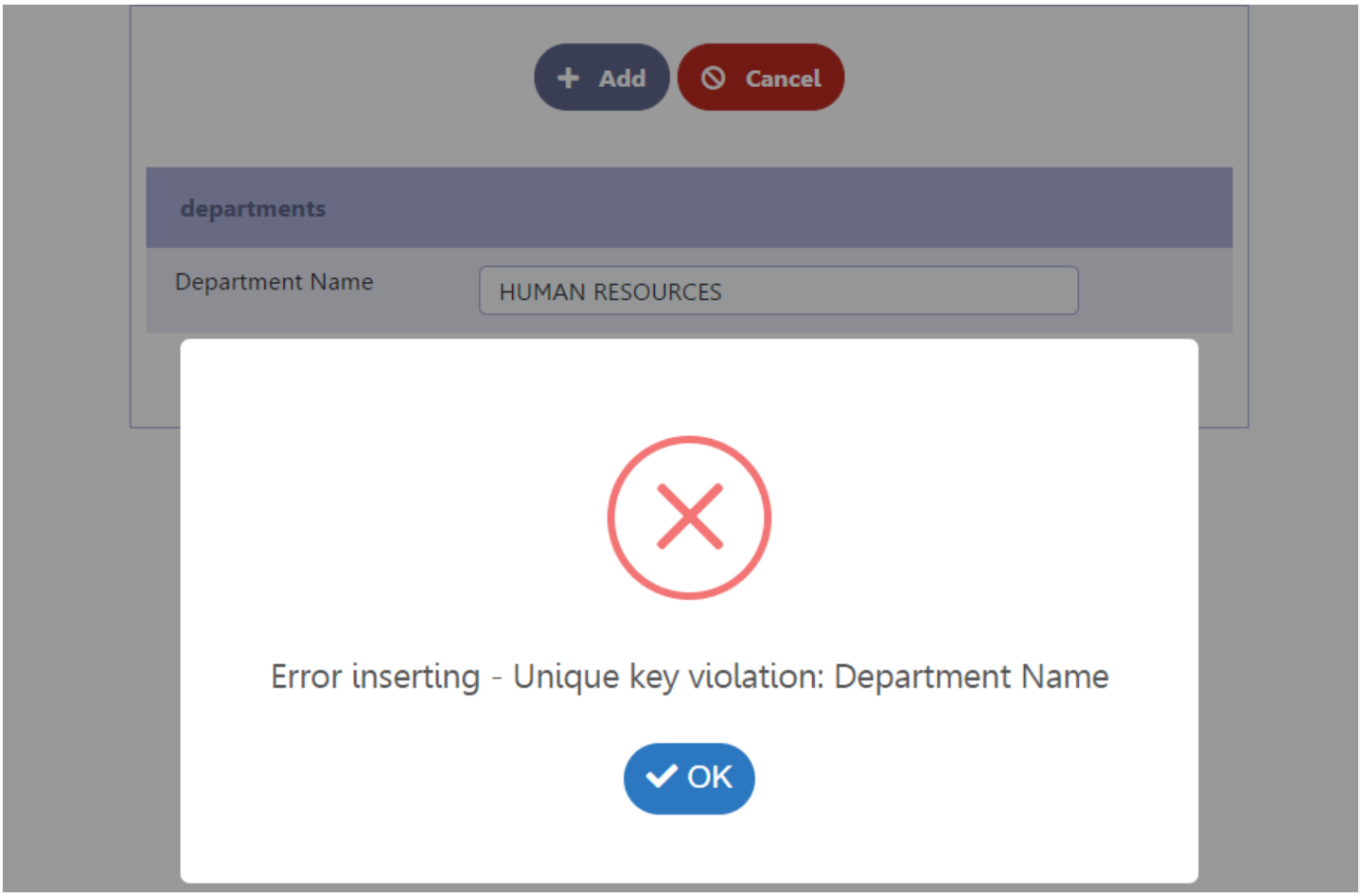
Unique key

With the unique key option you can inform which fields' data should not have repetitions within the database.

Unique key configuration Interface

If you want to combine two or more fields as unique, select the fields by holding the CTRL key of your keyboard and then click on the "Add" button.

The user will receive message if tries to insert a repeated value into the application. The validation and message are created automatically by Scriptcase, you are able to customize it by accessing the application language option and searching for the lang variable: **{lang_errm_inst_uniq}**. This lang variable contains a pre-set error message content.



Unique key message example

Related Links

Related Links 🔗

Related Videos ▷

Calendar Toolbar

The application toolbar has two segments: Top and Bottom, in a way that is possible to define to display buttons into both areas. Those areas work independently, allowing them to display the same button, for example.

It's also possible to select the buttons and their position if the application is running on a mobile device.

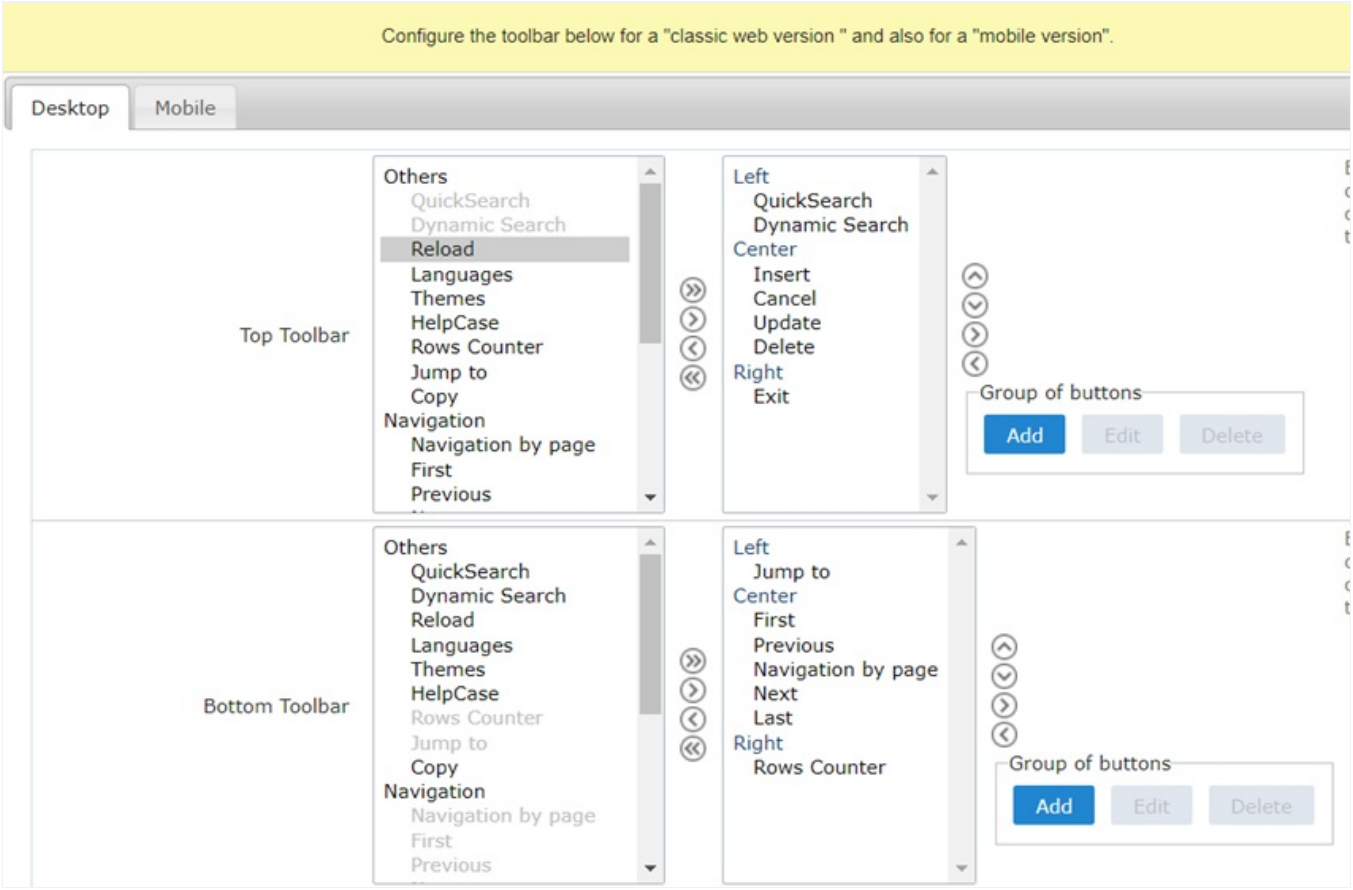
Toolbar

Desktop

Here we must inform the toolbar settings for the "Classic Web Version" mode and which buttons are available in the application when accessed from a **Desktop** environment.

Mobile

Here we must inform the toolbar settings for the "Mobile Version" mode. That is which buttons are available in the application when accessed from a **Mobile** dispositive.



Navigation:

Buttons relative to the navigation of the application.

Next	Move to the next page that can be a single record or a list of records.
Previous	Returns to displays the previous page records or a single record.
First	Move to the First page or record
Last	Move to the Last page or record
Exit	Close the application
Navigation by page	Displays a “page-number” navigation bar. **Example:** 1 2 3 4 5
Reload	Displays a button to reload the query data

Export:

The options available to export the Records. Scriptcase generates the following export formats for Forms:

PDF	Generates all the data of the application in a PDF format.
Print	Creates an HTML with the records ready for printing.

Update:

The CRUD options available in the Form.

Insert	Inserts the record into the database.
Update	Saves the changes made in a record.
Delete	Deletes the selected record.
Cancel	Cancel the changes made in a record before the insertion.

Others:

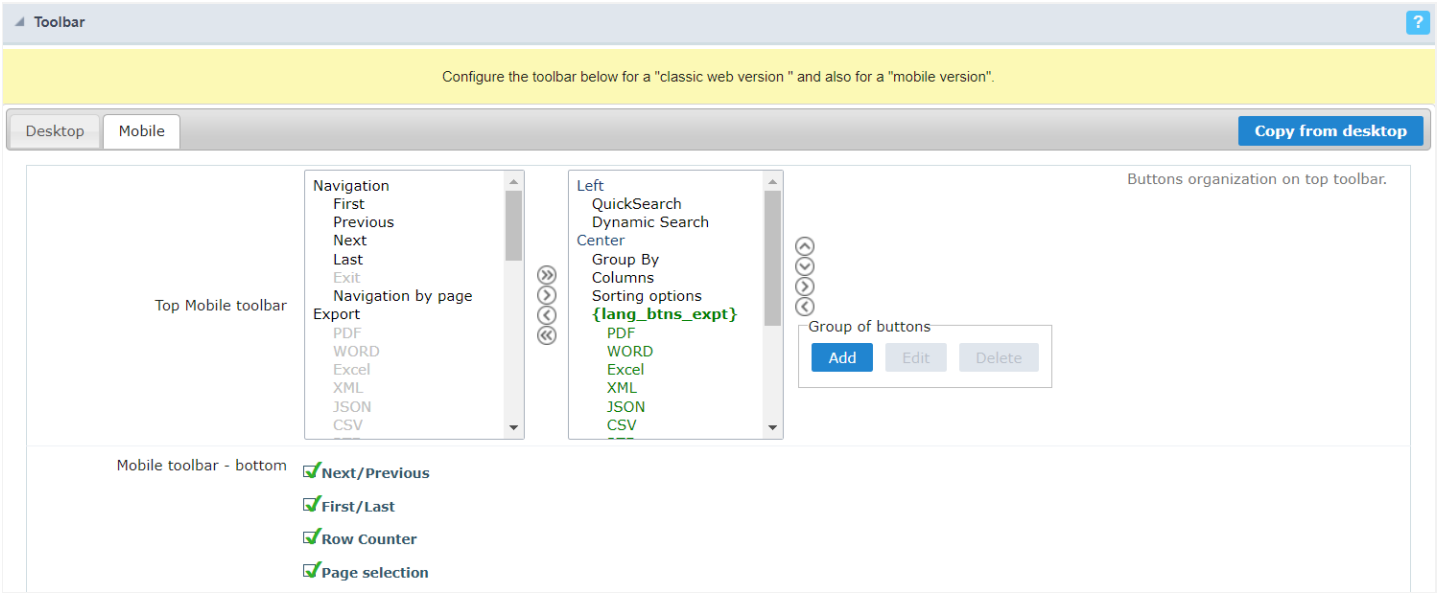
Other options available in the Form application.

Jump To	Move to the informed page or record.
Copy	Copy the current record data to another one.
Quick Search	Perform a quick search in the records of the application.
Dynamic Search	It displays the fields of the search to filter the records.
Languages	Displays a Combobox with the languages available in the project properties.
Themes	Displays a Combobox with the languages available in the project properties.
Rows Counter	Displays the number of records retrieved in the application.
HelpCase	Displays a button to open the help page.

Separator:

————————	Displays a line separating the buttons.

Toolbar Mobile



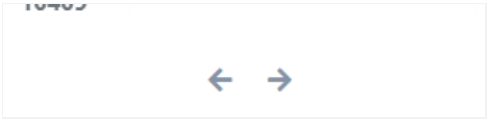
Top Mobile toolbar

It has the same options as the **Desktop** version, adding only the item “Copy from desktop”, which, when clicked, makes a copy of the items from the upper toolbar of **Desktop** to **Mobile**.

Mobile toolbar - bottom

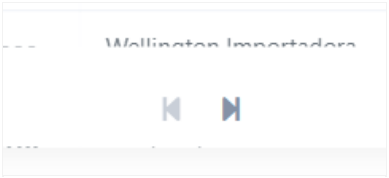
Next/Previous

Enables navigation to the next and previous page on mobile devices.



First/Last

Enables first and last page navigation on mobile devices.



Row Counter

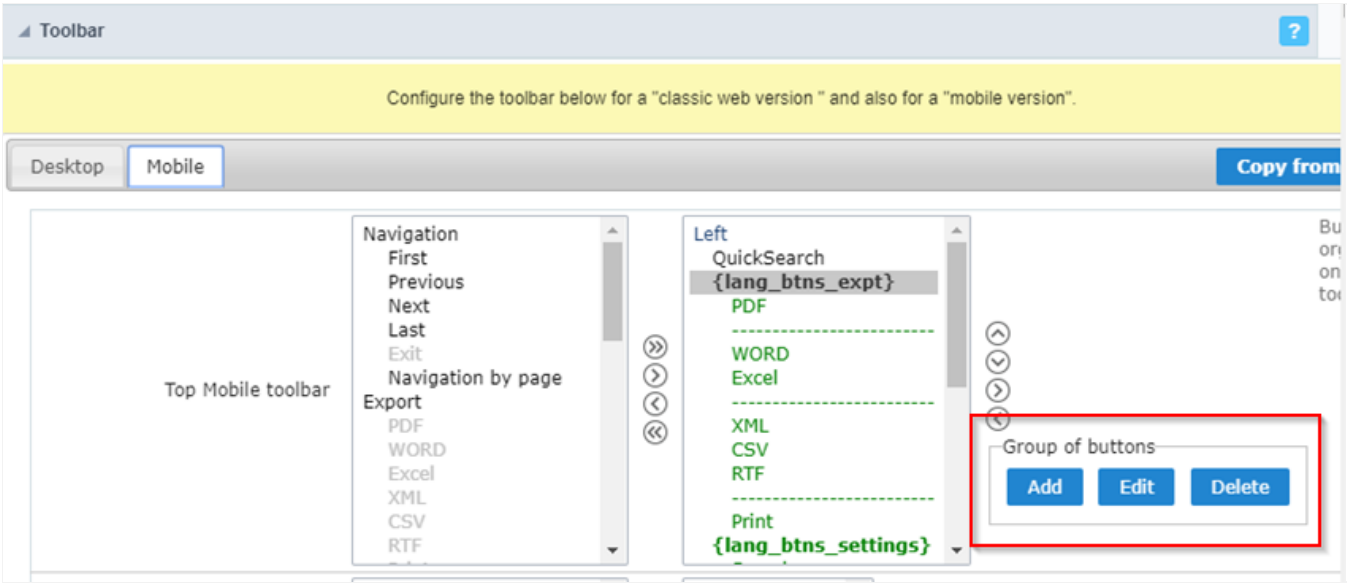
Enables the record counter showing the application's total records

Page selection

Enables page navigation on mobile.

Bottons Group

The **Group** option allows you to group a set of buttons of the application toolbar to display them as a dropdown, for example.



Add

Add a new group of buttons.

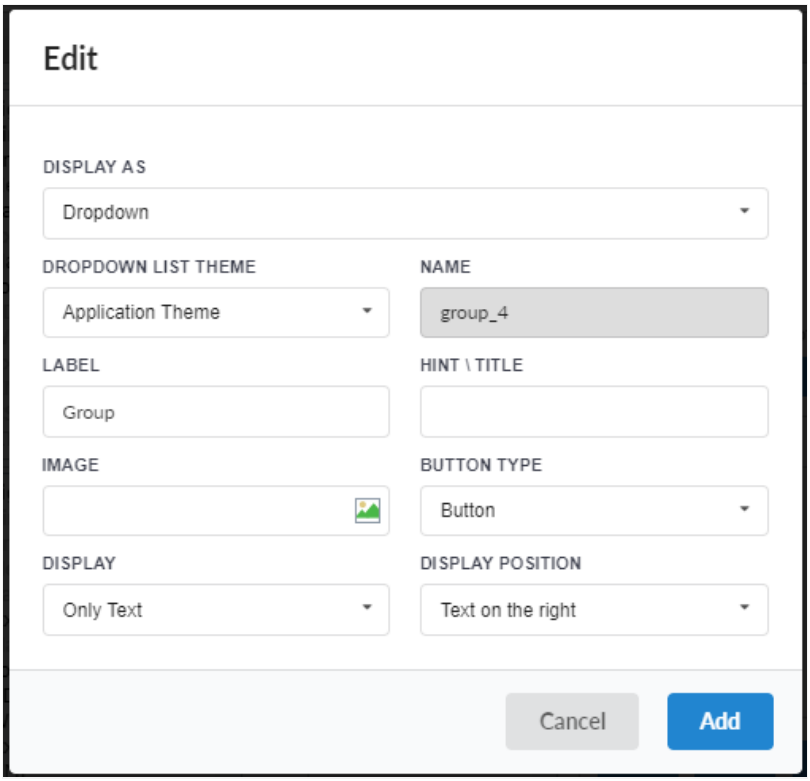
Edit

Edit an existent group of buttons.

Delete

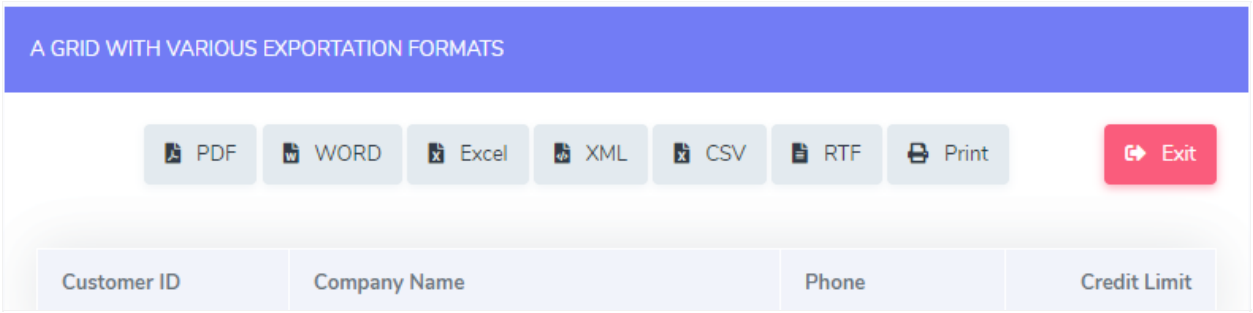
Delete the selected group of buttons.

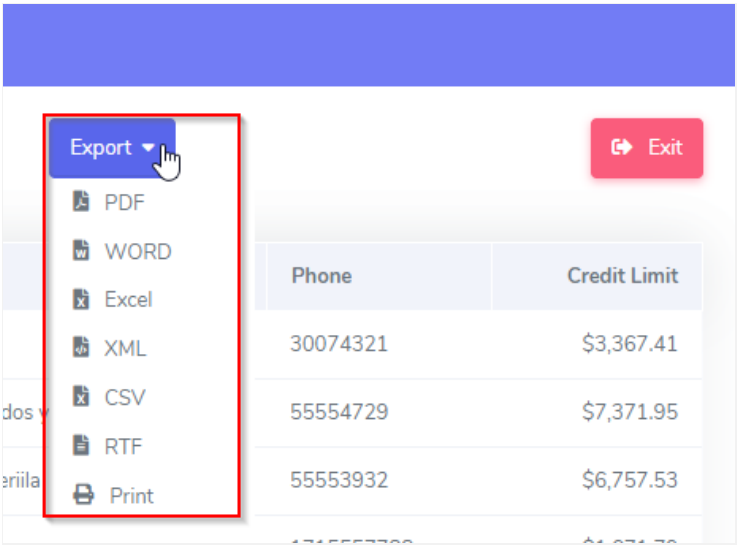
When you press the **Add** or **Edit** option, you can see the settings to configure the grouper:

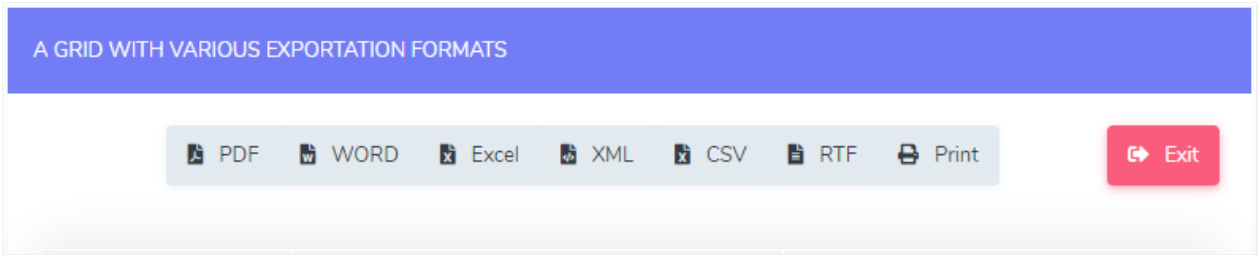


Display As

Allows displaying the group button as **Dropdown** or **Side by Side**.







Dropdown List Theme

Allows defining the Dropdown theme selecting between **Application theme** and **Button theme**.

Name

Allows defining a name for the button group.

Label

It is the displayed name for the button group in the application.

Hint\Title

Displays a hint to the end-user when the mouse is on the group of buttons.

Button Type

Allows displaying the button group as a Button, Image, or Link.

Image

Allows selecting an image for the button.

Display

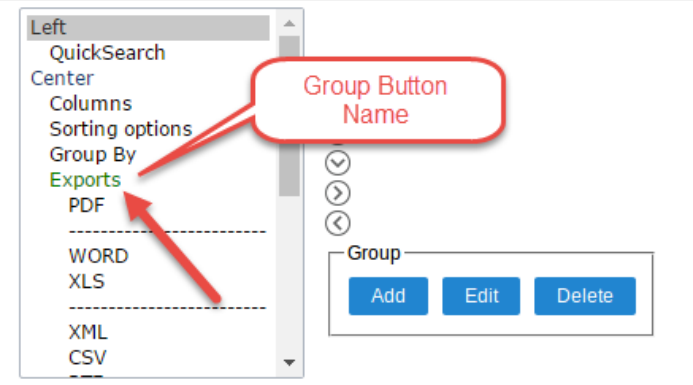
Defines if the button displays only Text, only image, or both.

Display Position

Defines the position of the Text or Image (Text to the right, Image to the right).

After creating a button group, you need to move the grouped buttons below of the Button Group and then move them to the

right. Like the image below:



Buttons Settings

Button	Label	Hint
QuickSearch		
Dynamic Search		
Insert		
Cancel		
Update		
Delete		
Exit		
Jump to		
First		
Previous		
Navigation by page		
Next		
Last		
Rows Counter		
New		

Button:

It displays the buttons available in the application.

Label:

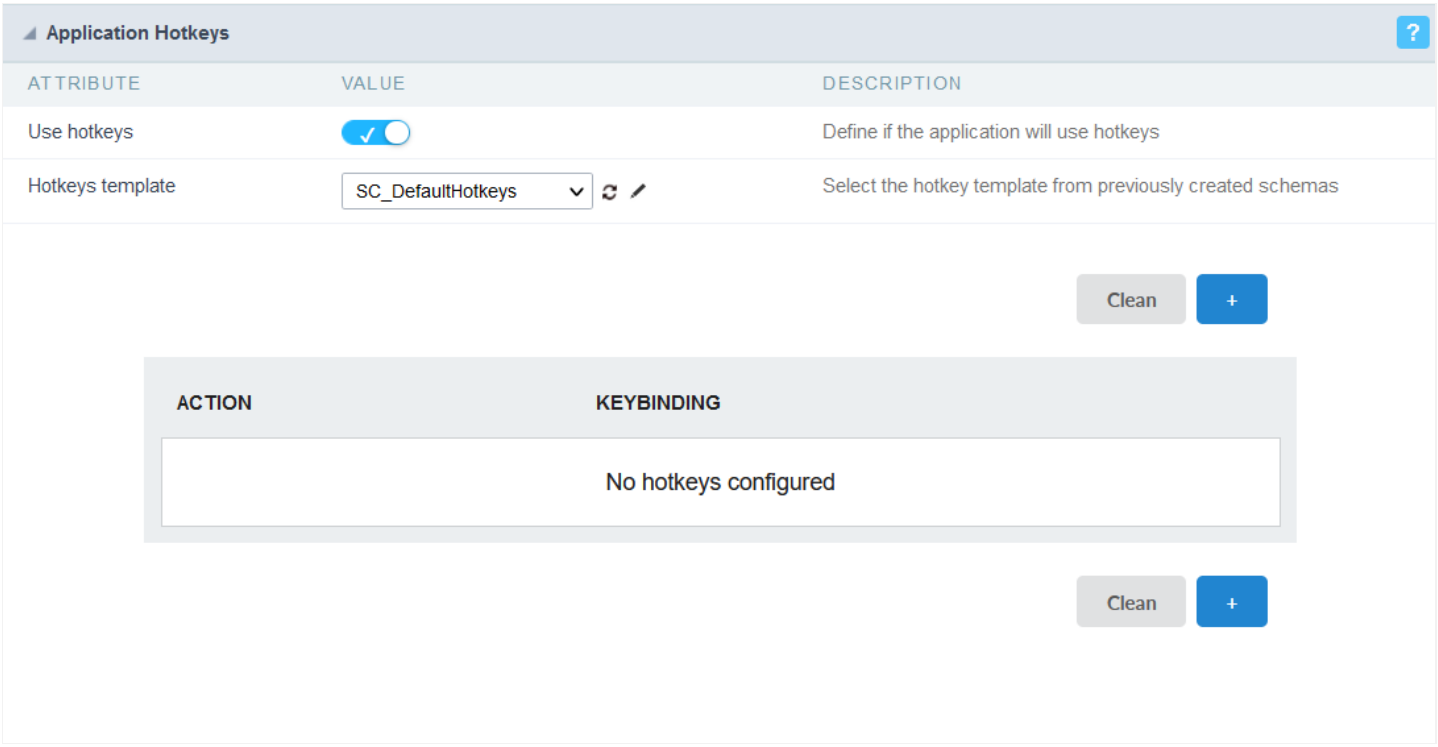
Allows defining the labels of the buttons to display for the users.

Hint:

Allows defining the buttons hint that to display for the users.

Application Hotkeys

Scriptcase allows creating shortcut keys to your applications. You can select a predefined template or create specific actions for an application.



Use hotkeys

Defines if the application uses hotkeys. When you enable this option, the default shortcut keys settings are disabled.

Hotkeys templat

Select the hotkey template previously created.

Action

Selects the triggered action when pressing the selected key.

Keybinding

Selects the keys responsible for executing the chosen action.

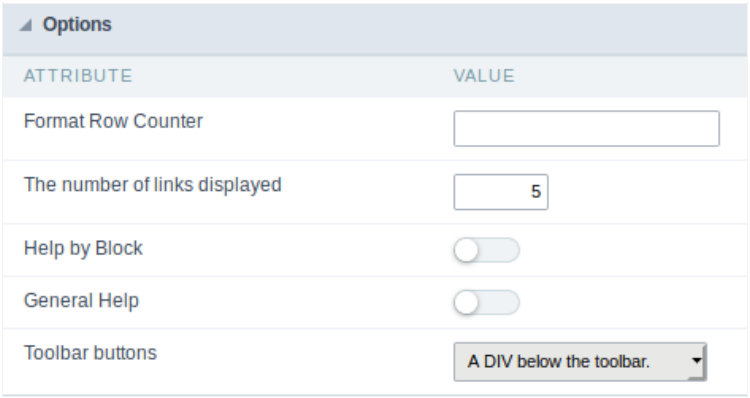
Add “+”

Adds a new action on the keys list.

Clear

It clears the selected hotkeys preference.

Options



Format Row Counter:

Allows defining the format of the row counter displayed on the application.

Example: (1 to 10 of 200)

The Number of Links Displayed:

Defines the number of links per page, when the navigation option is disabled.

Help by Block:

Indicates if the helps messages from relatives fields are grouped by block. In each field, we can define a help text. With this option activated, it shows up an icon in the block title bar to call the help page.

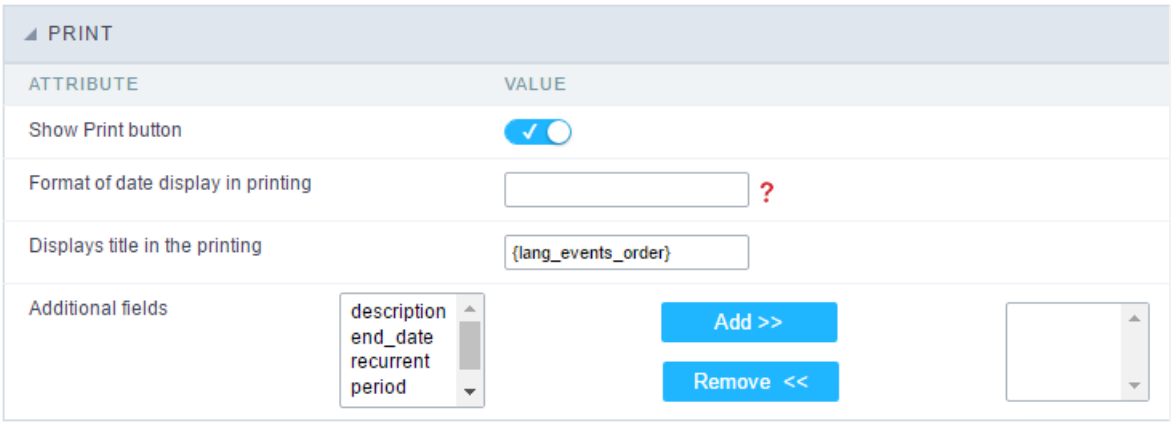
General Help:

The General Help “consolidates” all the fields help pages in a single page, putting an icon in the toolbar to call the help page.

Related Links 🔗

Related Videos ▷

Print settings



Print export settings.

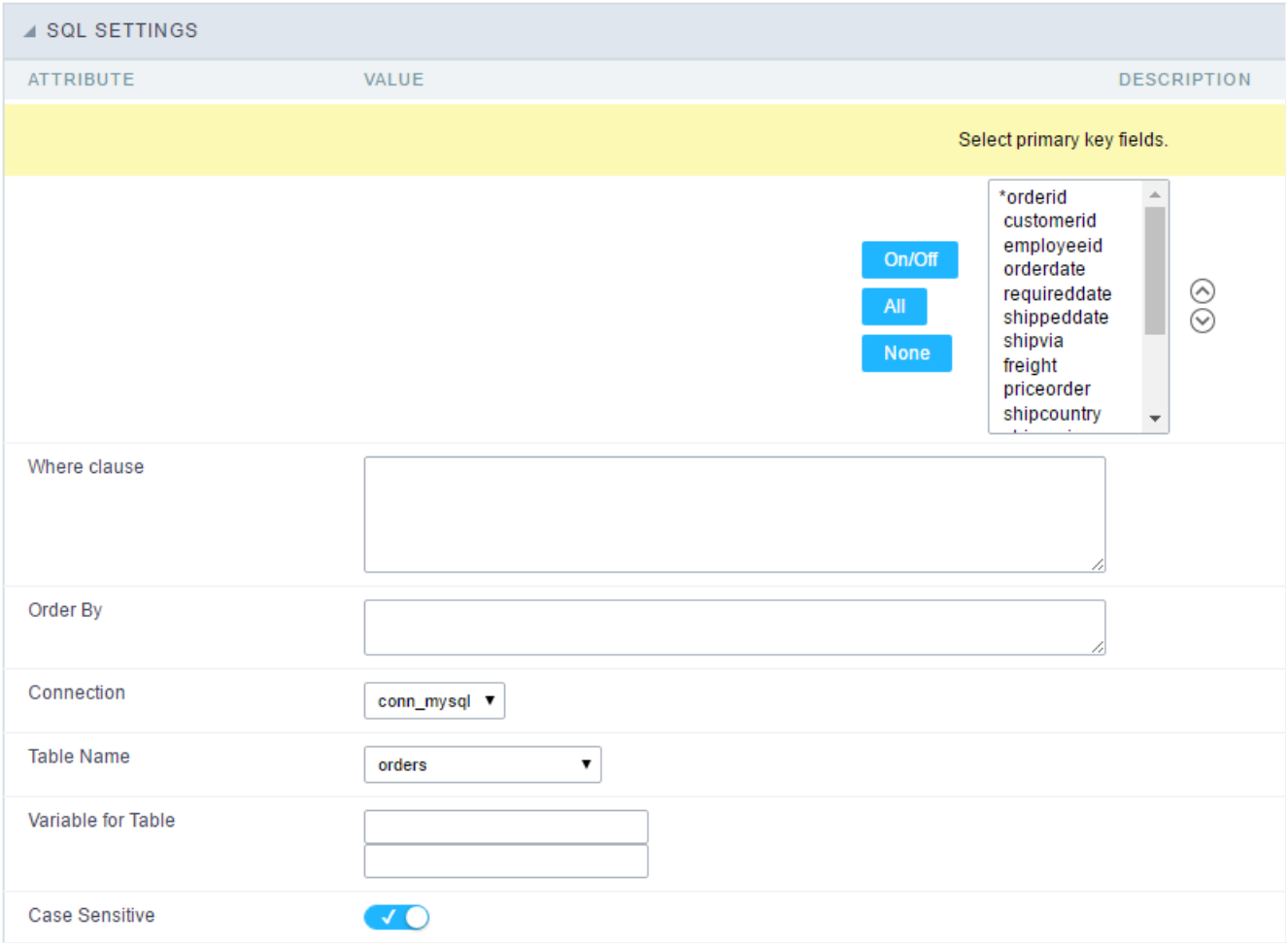
- **Display print button**: This option enable a button within the application to print the calendar.
- **Print format**: It changes the display data format for printing.
- **Display title for printing**: Displays the calendar title for printing.
- **Additional fields**: Additional fields to be displayed for printing.

Related Links

Related Videos

Calendar SQL Settings

This interface allows configuring the related database settings, such as the Primary Key, Filters, Sorts.



SQL configuration Interface.

Select primary key fields

It lets you define the Primary key of the Form. ScriptCase already identifies Primary Keys, but you can manually inform or change it by using the buttons beside the fields list. See how the buttons work:

- **On/Off** : Adds or Removes the attribute that defines the primary key for the field. The primary key fields have an asterisk beside their names.
- **All** : Defines all fields as Primary Keys.
- **None** : Defines none fields as Primary Keys.
- **Sorting Button** : These are the arrows on the right side of the Combo box. It allows ordering the fields of the Primary Key, placing it in the desired order. To order them, click on the field and use the arrows to move it.

Where clause

It allows adding a WHERE clause to filter the SQL records.

Order By

It allows adding an ORDER BY clause to determine the order to display the records. By default, it uses the primary key to sort the records.

Connection

It allows defining the database connection of the application. You can change the connection to another one that has the same table.

Table Name

It informs the database table used in the Form.

Variable for Table

It allows to use a variable to change a part of the string containing the table name.

Variable for Table Configuration.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the table you want to replace (replace from).

Case sensitive

It defines if the database connection uses case sensitive or not.

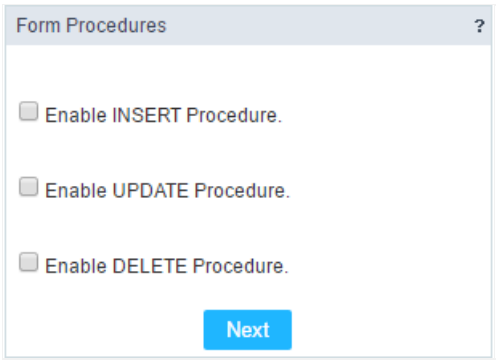
Related Links

Related Videos

Procedures

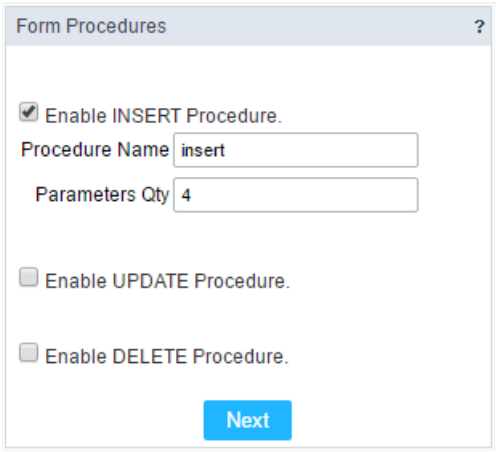
This interface allows configuring a Form application to execute the Stored Procedures from your database for Insert, Update and Delete records in the form. You don't need to use the three options simultaneously. If a Procedure option is not enabled, it continues to work as default by using the INSERT, UPDATE and DELETE commands.

See below, the step-by-step to enable a Stored Procedure for inserting records into the database.



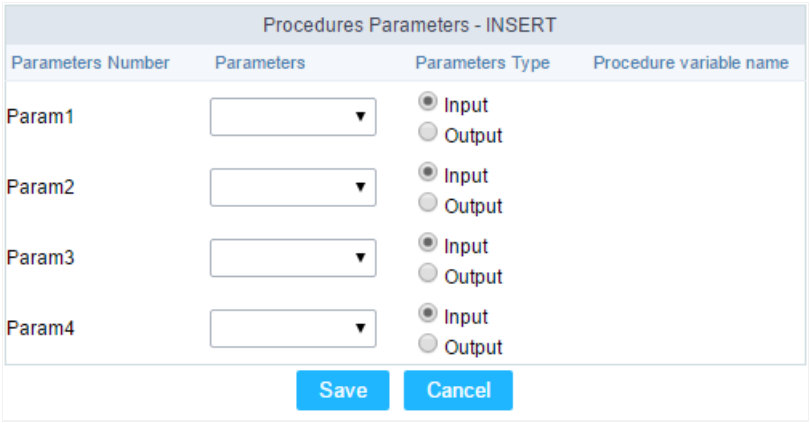
Form Stored Procedures Interface.

- Let's check the first option: "Enable INSERT Procedure". Then, inform the Name of the Procedure and the number of parameters.



Defining parameters for the Procedure.

- Now associate the fields for each parameter of the Stored Procedure and the type of parameter (Input or Output). Click on the "Save" button to finish.



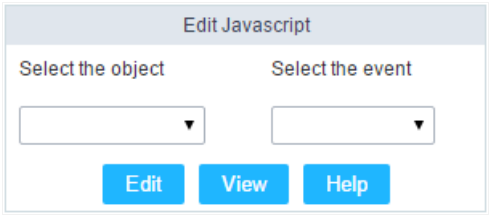
Passing the values to the Stored Procedure Parameters.

Related Links

Related Videos ▷

JavaScript

To coding with JavaScript in Scriptcase Form application, we must associate a JavaScript event to a form field.



Edit JavaScript Interface.

Select the object

This Combo box displays the fields of the form application, and also the form itself as an object

Select the event

Use it to associate the event with the selected field to apply the JS code. View the events available:

- **onclick**: Acts when clicking on the field.
- **onblur**: Acts when the focus leaves the object.
- **onChange**: Acts when the focus leaves the object, and there are changes in the value.
- **onFocus**: Runs when the focus gets in the object.
- **onMouseOver**: Runs when the mouse pointer hovers the object.
- **onMouseOut**: Runs when the mouse pointer moves out the object.

Events related to the Form

The events below are associated directly with the form object.

- **OnLoad**: This event runs when the page is loaded, also when clicking on the navigating buttons.
- **onSubmit**: This event runs when clicking on the “New”, “Save”, and “Delete” buttons.

Edit JavaScript

*Select the object and event, then click on the Edit button. It opens a page to inform custom JavaScript routines and standar

![Edit JavaScript Interface][javascript_edicao_code]
Edit JavaScript Interface

> The JavaScript language doesn't have the same behavior in all the browsers available. A tip is to test running the applicatic

- **OnClick Example**
  - When clicking on a field of type radio, you can enable or disable form fields according to the selected value.

```
if(document.F1.gender[0].checked){
   document.F1.maternity.disabled = false;
   document.F1.maternity.style.background='FFFFFF'
}
if(document.F1.gender[1].checked){
  document.F1.maternity.disabled = true;
  document.F1.maternity.style.background='FCEEBC'
}
```

To access the values of a radio field, you need to use the index.

- **OnBlur Example**

- You can define a warning for the field “weekly_work_time” when the focus is getting out it.

```
if (document.F1.tp_point[0].checked  && document.F1.weekly_work_time.value > '20')
{
    alert("The work time exceeds the limit allowed");
    document.F1.weekly_work_time.value = "";
    document.F1.weekly_work_time.focus();
}
```

- **OnChange Example**
  - By modifying the “Salary” of an employee and leaving the field, we’ll check if his “position” is “gardener”.

```
if (document.F1.salary.value > 2000.00 && document.F1.position.value == 'gardener'){

alert('When I grow up, I want to be a gardener);
}
```

- **OnFocus Example**
  - After informing the purchase value and selecting the payment method in a Select object “Select: pay_method”, the JavaScript code below calculates the value of the purchase.

```
if (document.F1.pay_method[document.F1.pay_method.selectedIndex].text == 'Money')
{
     document.F1.total.value = document.F1.pauchase_value.value;
}
```

- **OnMouseOver Example**
  - You can change the style (background color, font and font color) when the mouse hovers the field.

```
document.F1.field_name.style.bgColor = "0FFCCA"
```

- **onMouseOut Example**
  - Sets the background color when the mouse moves out from the field.

```
document.F1.field_name.style.bgColor = "FFFFFF"
```

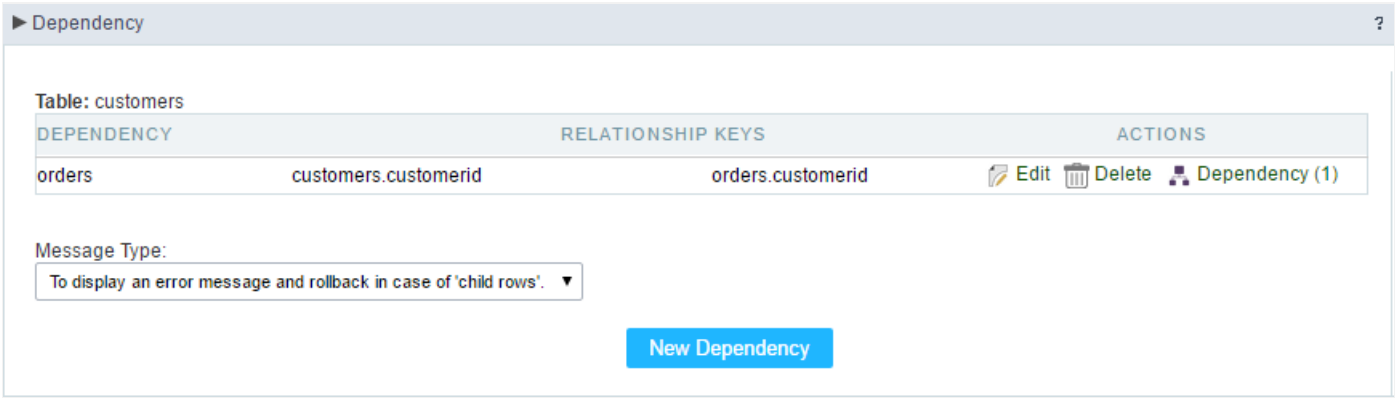
Related Links 🔗

Related Videos ▷

Dependencies

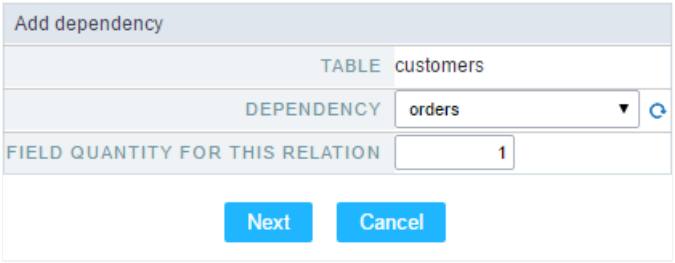
This feature allows linking the tables that contain relationships, dependencies, like the tables orders and order_details. This way, it's possible to delete a record in the table orders, and it automatically deletes all the details for that order as well. Below let's see a practical example of this feature.

1. Firstly, you must define a new dependency. Then you must select the dependent table. Click on the button New Dependency to start.



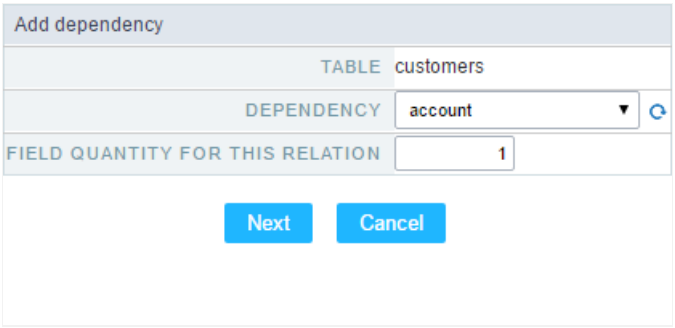
Creating a new Dependency.

1. You must inform the fields amount is related between both tables. In this example, we have only one field.



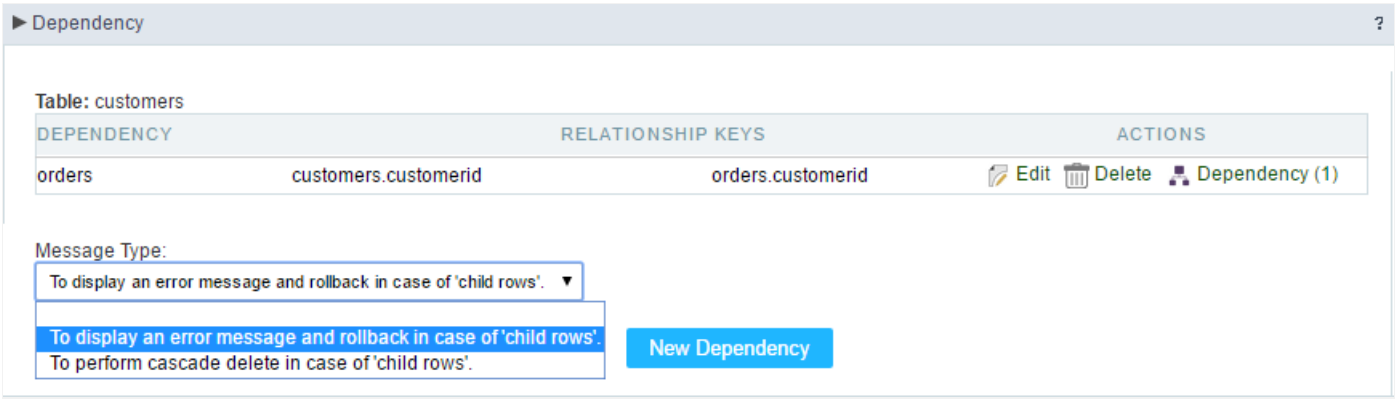
Dependency Table configuration.

1. Then select the related key fields: order_details->Orderid - orders->Orderid.



Dependency Key Fields Configurations.

1. Now we need to select behaviour for the application. Then click on the button Generate Scripts to define the Dependency Rule.



Configuring dependency Rules.

Related Links

Related Videos

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application.

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Enable CSRF

With this option enabled, the scriptcase prevents a malicious attack on a page where unauthorized commands are transmitted through a user the page trusts.

These attacks are known as a “Cross-Site Request Forgery” attack.

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Headers

Disable XSS Auditor

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Captcha

Captcha

Use Captcha

Defines if the application uses Captcha.

Number of Characters

Amount of characters in the Captcha image.

Character List

List of characters used in the Captcha.

Label

The message displayed for the Captcha.

Error message

Captcha error message.

Height

Height of the Captha image. (in pixels)

Width

Width of the Captha image. (in pixels)

Font Size

Font Size of the Captha image. (in pixels)

Reload

Display the refresh button in the Captcha.

Select one layout

It offers layouts for display the Captcha.

Recaptcha

ReCAPTCHA is an API provided by Google for forms. It adds security, preventing automatic submission of forms through robots.

reCAPTCHA sample:

1. First, we must request an API Key to activate reCAPTCHA into a Scriptcase application by following the steps below:

   To get a **Site key** and **Secret Key** go to the link: https://www.google.com/recaptcha/admin#list.
   See the image:

Label

It is a project identifier to create the reCAPTCHA keys.

Choose the type of reCaptcha

We must choose the option **reCAPTCHA V2**.

Domains

We can insert multiple domains (one per line) to limit the API uses.

1. Then, we need to accept the Terms of Service (“Accept the reCAPTCHA Terms of Service”).
2. When clicking on **Register**, the page refreshes and shows the integration of reCAPTCHA information. There we can get the **Site Key** and **Secret Key**:

1. Now, we can set the Scriptcase application security:

Site Key

Key generated by Google after reCAPTCHA project creation.

Secret Key

Key generated by Google after reCAPTCHA project creation.

Theme

Defines the reCAPTCHA color. There are two options:

- **Light** :
- **Dark** :

Type

The type of reCAPTCHA. There are two options:

- **Audio**:
- **Image**:

Size

The size of reCAPTCHA. There are two options:

- **Normal** :
- **Compact**:

Position

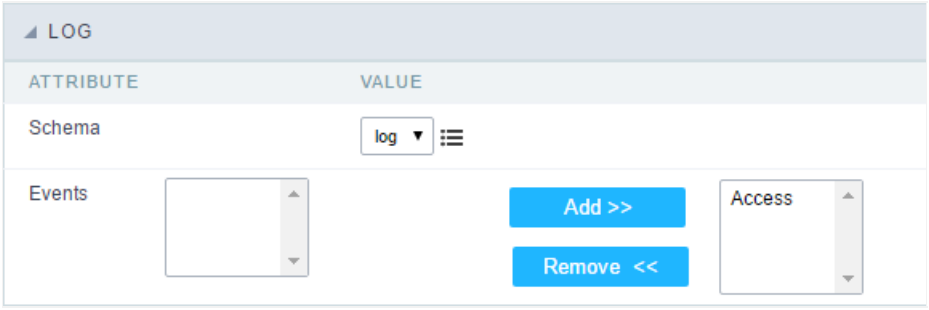
Here we can define the reCAPTCHA component alignment:

- **Left**: Position the reCAPTCHA component at the left.
- **Center**: Position the reCAPTCHA component at the center.
- **Right** : Position the reCAPTCHA component at the right.

Calendar Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

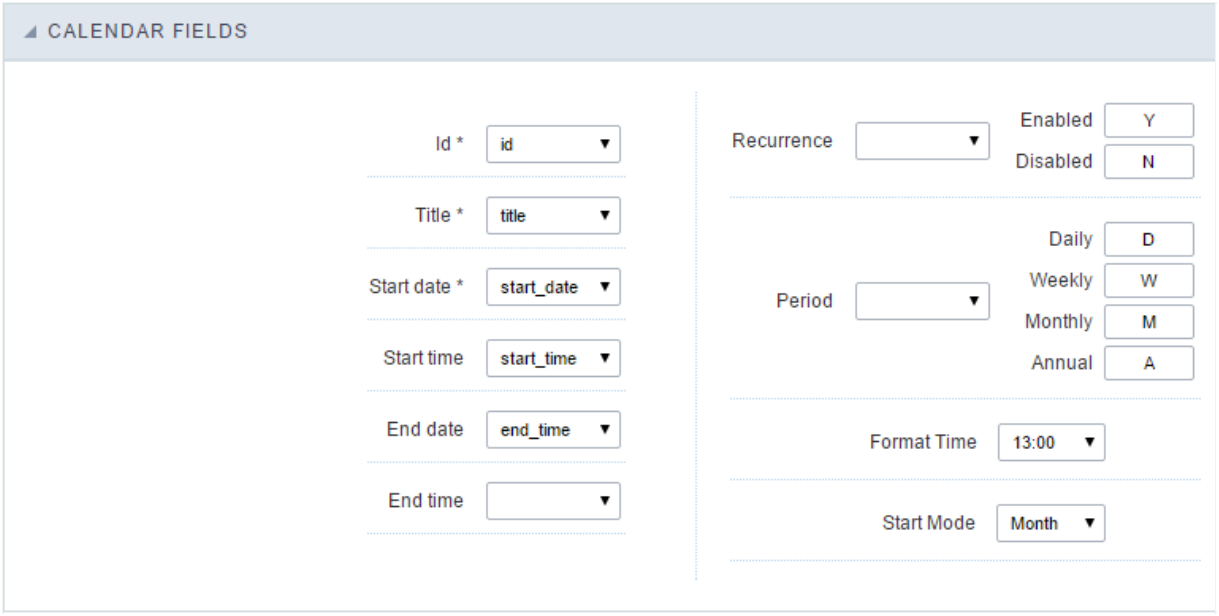
Related Links

Related Videos

Calendar

Calendar fields

This interface displays the basic Calendar field settings.



Calendar fields interface.

Related Links

Related Videos

Calendar - Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid		
Procedure		
Chart		
Form		
Control		
Calendar		
Search		
PDF Report		

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven't manage fields.

Calendar - Text Field

General Settings

This type of field allows the developer to create quickly inputs to insert and update data, where the final user can inform its data to be allocated in its database.

Type of Data

Defines the type of the field in the application. In this case we should select **Text**.

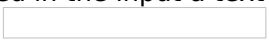
Label

Defines the title that will be displayed in the field when the application is executed. The terminology used in the interface is fundamental to the system have a good usability, whe should use names and familiar terms to the final user of the application, instead of using terms from the system.

For example, this text field that has the name **customerid**, the client would have a much better understanding of the functionality of the field when we define the label as **Customer Name**.

Not only a fixed text, the **Label** attribute allows the use of langs to define the field's title, making it possible to use your application in a multi language project.

Watermark

Informing a text to the **Watermark** it will be displayed in the input a text as an example that can be informed in the field. The result after set will be this: 

Initial Value(type)

Allows the initial definition to the field when the form is in insert mode. It is possible to choose between two options:

Defined Value: When this option is selected, the Initial Value attribute will be available, where we should inform the field's initial value. For example, my initial value is **Arlindo**, when a new register is inserted, the field **Seller Name** will be initialized as Arlindo.

System Data: When this option is selected, the initial value will be the actual date of your computer's system.

Amount of Characters

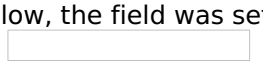
Allows to set the width of the text field's input that varies with the amount of characters informed. Although, if the amount of characters typed are greater than the setting, the text will be pushed to the left, to keep the maximum amount of characters as defined.

Show HTML Content

When this option is active every HTML, CSS and JavaScript content that are in the database will be displayed with the main value.

Validation Image

When this option is active, a image will be displayed next to the informed field if the field is according to the settings defined by the developer.

In the example below, the field was set to receive at least 5 characters, see what happens when informed only 4 characters: 

However if informed 5 or more characters the field will be displayed as:

Password Field

When this option is active, the text field will be converted to the format used in password fields. For example:

View password characters

By enabling this option, a button will be displayed in the password field so that the password is displayed when clicked.

Save Variable

Allows to save a session variable(global variable) with the field's value, to be used in others applications.

For example, in the login form the username can be saved in session and displayed on the header of others applications.

Variable Name

In this attribute we should define the name of the session variable, active in the previous item, that will receive the field's value.

We should inform only the variable's name, - **var_rating**.

The method to use its value is global variable.

Field Mask

Defines the field mask. There are two types of mask described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

For example, it is possible to set the mast to display a telephone number:

It will be show with this format on runtime:

It is also possible to set the field mask like those examples:

Field mask examples:

Telephone number

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Validate with mask

When this option is active it is possible to insert the data with the mask set in the **Field Mask** option.

Complete to the left

When this option is used, the value defined will be added on the left of the information inserted in the field if the value is lower than the maximum set in the **Maximum Size** option.

Field size in database

Defines the field's size related to the size set in the database. This value is already set automatically by default when the application is generated.

Hidden Field

When this option is active, the field will be hidden in the application on runtime.

Label Field

When this option is active, the field will be altered to only a label where the info will be displayed, where updates or inserts will not be possible.

Save HTML tags

When this option is active, it allows to HTML tags in the field to be inserted with the data, instead of being interpreted.

Text input in JavaScript

When this option is active, it will be show every JavaScript content inserted in the database with its main value.

> This option can only be used using the **Editable grid view**.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disable Field

Allows the developer to disable the field, so the user can not type a value according with the option defined by the developer.

The available options are: **No** - Initial value of the attribute, this option does not disable the field.
Update Mode - This option only disables the field when editing existing registers.
Insert Mode - This option only disables the field when inserting new registers.
Insert / Update Mode - This option disables the field when editing exciting registers or when inserting new registers.

HTML Type

HTML object used to display the field in the form.

SQL Type

Informs the type of the field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

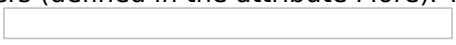
Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed. 

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font's size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.

- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Multiple Lines Text Field

General Settings

Multiple Lines Text field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Multiple Lines Text , you can inform a Text value to the field in multiple lines.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : you will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Define the amount of characters allowed for the field.
- **Validation Image** : Allows to display an image next to the field when it is being validated.
- **Lines** : Allows to define the amount of lines that the field will have at start.
- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the field size in bytes. it is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Save HTML tags** : This option allows to save the HTML content of the field to the database.
- **Text input in JavaScript** : Allows to save JavaScript code informed in the field to the database.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed.

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

```
SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
```

The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position

(left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Integer Field

General Settings

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Label:

It sets the title that will be displayed in the field when you run the application. The terminology used in the interface is fundamental for your system has a good usability. In this case, we recommend you to use names and terms familiar to the end user of the application, instead of using terms originated in the system.

Watermark

Informing a text for the **watermark** it will be displayed on the input text with an example of what can be reported in the field. The result after that will be configured:

Initial value (type)

Allows setting an initial value for the field when the application is in inclusion. You can choose between two options:

Pre-set value: When you select this option, the Initial Value attribute will be made available in the place where we inform the initial value of the field.
For example, my initial value is **Arlindo** when inserting a new record the **seller_name** field will start with Arlindo.

System date: When you select this option, the initial value will be the current system date of your computer.

> If you select the type system date, it will be not necessary to fill in the initial value attribute.

Amount of characters

It allows you to set the width of the input text field according to the amount of characters. However, if the quantity entered is greater than the set for the characters, the text will be pushed to the left, in order to ensure the maximum amount of characters set in the option of **Values formatting**.

Validation Image

When you enable this option, an image will be shown next to the field informing whether the field is in accordance with the settings of *minimum size* and *Maximum* size (as you can see in the images below) defined in the Formatting value option.

In the example below, the field was set to receive at least 5 characters, see what happens with the image when it receives only 4 characters:

However, if the value inserted has 5 or more characters the the image will change according to example bellow:

Use slider:

It displays a slider component in the field. So you can increase or decrease the value sliding the cursor. You can also customize the increment value, if it increments the value 1 by 1, 2, 5, 10... N.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Validate with Mask

By enabling this option you will be able to enter data according to the mask that was configured in the option **Field Mask** and Scriptcase will validate it.

Record Variable

Allows you to record a session variable with the value of the field ([global variable] [var_glob]), to be used in other applications.

Example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable Name

In this option you must set the session variable name, enabled in the previous item, which will receive the value of the field.

You need to inform only the variable name, for example: **var_seller**.

The recovery of the value is made as [Global Variable][var_glob]{:target='blank'}.

Field size in Database

This option sets the size of the field relative to the size that is configured in the database. By default this value is already configured automatically when the application is generated.

Hidden Field

This option when enabled will hide the field inside the application at the time of execution.

Label Field

By enabling this option, the field will be changed to only one label where the information will be displayed, so it is not possible to make changes or inserts in the field configured as label.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disable field

Allows you to disable the field, making it impossible for the user to enter a value according to the developer-defined option.

The available options are:

No - initial value of the attribute, this option does not disable the field. **Update mode** - This option disables the field only when editing the records. **Insert mode** - This option disables the field only inserting new records. **Update/Insert mode** - This option disables the field in both editing and inserting new records.

HTML Type

HTML object used to display the field in the application.

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Values Format

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Digit grouping:

It allows you to define if the field displays the digit grouping separator.

Maximum Size:

It allows you to define the maximum size that will be used in the field of the application.

Minimum Value:

It allows you to define the minimum value that will be used in the field of the application.

Maximum Value:

It allows you to define the maximum value that will be used in the field of the application.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only

available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Accept:

It allows to define if the field accept only negative values, positive-only or both.

Display Calculator:

This option when enabled will display a calculator to assist the end user to perform calculations.

Use Spin:

This option applies a Spin component to the number field. So the end user can increase or increase the numbers using it. You will also be able to set a minor or major range for the spin.

Smaller increment for the Spin::

Sets the minimum range of the Spin component.

Greater increment for the Spin:

Sets the maximum range of the Spin.

Database Value

It allows you to define how the value to be written to the database.

Preset value:

It allows you to specify a fixed value or associate it to global variables created inside the project and save it automatically to the database field.

Auto increment (automatic):

It allows you to use the increase generated automatically by the database to fill the field. It is used only when the database field is of type AUTO INCREMENT or similar. For the databases that use string, such as Oracle, PostGres and Firebird, you must inform the name of the string.

Auto increment (manual):

The application itself will simulate an auto increment in the field. So the value of the field will be calculated automatically by the application.

Date of inclusion:

The field will be populated automatically with server date during the inclusion.

Date of update:

The field will be populated automatically with server date during the update.

User IP:

The field will get the IP of the machine that is accessing the application.

Calculated by the database:

The field value is assigned by the database itself. The field will not be used in the process of updating. For example: it will be updated via a Trigger.

Null:

The field will be automatically filled with null.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:

Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

```
SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}
```

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1

Label	Value	Start	Size
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Display Settings

You can set CSS values for the field individually. Thus, for example, you can highlight one field of the others in the application. As CSS field properties, when changed, they are added to a class created automatically by ScriptCase for each application field.

> Individual field CSS settings, when inserted, override theme settings (CSS of applications (Themes)) selected for application.

The settings are divided into three property blocks, these blocks are:

CSS of Title

This makes it possible to change the CSS properties of the field's Label.

In the example below, You can see the difference of field title configurations. While the field **Contactname** have the same formatting, inherits the theme of the application, the field **Contacttitle** have a different formation of of others, from the changes made in the CSS of the field.

CSS of field

Changes the CSS properties of the <td> where the input object (where the user enters data for insertion into a form) is positioned. In the image below, you can see where the change is applied.

Field Contacttitle with changes to Field CSS properties, changing background color and

horizontal alignment

CSS of Input object

Changes the CSS properties in the Input of field object, where the user type the data in a form.

Field with changes to CSS properties of Input Object, changing background color and input source color

CSS properties

The available configuration options are basically the same for each of the configuration blocks above.

description of available configuration attributes

Font

Changes the font of the text according to the fonts selected using the *font-family* property.

In this option, some types of fonts are provided to you (as shown below).

Font Size

Changes the font size of the text using the property *font-size* in the field class.

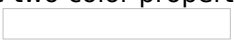
You need to select the available value from our list, the measure used for this property is the pixel.

Font Color

Define the font color used by the property *color* in the field class.

Background color

Define the background color using the property *background-color* in the field class.

The colors that will be used in the two color properties listed above, **Font color** and **Background color**, can be entered via the color palette -  - available next to the field or manually entered values in the accepted formats that are: *Hexadecimal*, *RGB*, *RGBA*, *HSL*, *HSLA* or o *Color name*.

Color palette
By clicking on the color palette icon - - next to the field, a window will open with some default colors.

When you select one of the colors, a value in hexadecimal format (HEX) will be entered, representing the chosen color.

Hexadecimal
Acronym for hexadecimal, this code is composed of the pound sign (#) plus six digits. The first two define the intensity of the color red, the middle two are green and the last two are blue.

Bold

Applies bold style to the font.

Underline

Lets you apply the underlined style to the font.

Border style

Defines the border font style.

Border Collapse

Defines the border collapse.

Border size

Changes the size of the title border.

Border color

Chooses the border color , using a color palette to apply to the title.

Horizontal alignment

Position the filter label at desired location (left,rigth,center e justify).

Vertical alignment

Position the label of the filter in the wanted location baseline, sub, super, top, text-top, middle, bottom, text-bottom).

Not available in Input Object CSS

Width

To set the width of the title.

Height

To set the height of the title.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be

displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Decimal Field

General Settings

Decimal field configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser’s autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser’s autocomplete.
- When **enabled**, the field’s autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit grouping

It defines if the field displays the digits separator.

Maximum Size

It defines the max size of the field.

Minimum Value

It defines the min value of the field.

Maximum Value

It defines the max value of the field.

Decimal Precision

It defines the number of decimal places for the field.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Accept

It determines if the field accepts only negative, positive, or both numbers.

Show a calculator

It displays an icon beside the field to help the user calculating the field value.

Auto-complete with zeros

It automatically informs the decimal places when not specified. If disabled, the user always needs to inform the decimal value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected

below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Currency Field

General Settings

Currency field configuration Interface.

Data Type

Define the field type to Currency. It allows defining the currency number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser’s autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser’s autocomplete.
- When **enabled**, the field’s autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit grouping

It defines if the field displays the digits separator.

Currency symbol usage

It defines if the field displays the Currency Symbol of the Regional Settings.

Maximum Size

It defines the max size of the field.

Minimum Value

It defines the min value of the field.

Maximum Value

It defines the max value of the field.

Decimal Precision

It defines the number of decimal places for the field.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Accept

It determines if the field accepts only negative, positive, or both numbers.

Show a calculator

It displays an icon beside the field to help the user calculating the field value.

Auto-complete with zeros

It automatically informs the decimal places when not specified. If disabled, the user always needs to inform the decimal value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The

select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Date Field

General Settings

Date field configuration Interface.

Data Type

Define the type of field. When setting it to Date, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server Date.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in "Insert Mode", "Update Mode" or in "Insert and Update Mode".

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Date separator

It allows you to inform the separator symbol for the date.

First Day

Defines the first day of the week.

Display

It offers predefined formats for displaying dates.

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

Use Combo-box

It allows you to select the date from a combo-box.

Minimum Date

- **Fixed Date**: Inform the minimum date allowed presented in the selected date format.
- **Current Date**: By clicking on the icon next to the text field, ScriptCase offers the following options:
  1. **Actual date**: Set the current date as the minimum allowed date.
  2. **Actual date with increment**: Set the minimum date with the current date more(+) the days, months, or years you want to increment.
  3. **Actual date with decrement**: Set the minimum date with the current date minus(-) the days, months, or years you want to decrement.

Maximum Date

- **Fixed Date**: Inform the maximum date allowed presented in the selected date format.
- **Current Date**: By clicking on the icon next to the text field, ScriptCase offers the following options:
  1. **Actual date**: Set the current date as the maximum allowed date.
  2. **Actual date with increment**: Set the maximum date with the current date more(+) the days, months, or years you want to increment.

3. **Actual date with decrement**: Set the maximum date with the current date minus(-) the days, months, or years you want to decrement.

Display Format

It enables the Date format beside the field when informing the date.

Date display position

Available when activating the **Display Format** flag, defines where the filling mode will be displayed. The options are: **Right**, **B Lower** and **Watermark**.

Display Calendar

It enables a calendar icon beside the field that allows selecting the date from a calendar to fill the input.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Time Field

General Settings

Time field configuration Interface.

Data Type

Define the type of field. When setting it to Time, you can inform a time format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser's autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser's autocomplete.
- When **enabled**, the field's autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in "Insert Mode", "Update Mode" or in "Insert and Update Mode".

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the Time. When not enabled, it displays the Time separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying Time.

Minimum time

Minimum value accepted by the field.

- **Fixed Time** : Enter the minimum time that scriptcase will criticize in the time type field in the format as presented.
    - **Current Time** : When clicking on the icon next to the box, scriptcase provides the following options:
    - **Current Time with Increment** : The minimum time will be the current time (+) the days or months or years that you increment.
    - **Current Time with Decrement** : The minimum time will be the current time (-) the days or months or years that you want to decrement.

Maximum time

Maximum value accepted by the field.

- **Fixed Time**: Manually enter the maximum time that scriptcase will input in the time type field in the format as it is presented.
    - **Current Time**: When you click the icon next to the box, scriptcase provides the following options:
    - **Current Time with Increment**: The maximum time will be the current time (+) the days or months or years that you increment.
    - **Current Time with Decrement**: The maximum time will be the current time (-) the days or months or years that you want to decrement.

Display Format

It enables the Time format beside the field when informing the time.

Display position

Available when activating the **Display Format** flag, defines where the filling mode will be displayed. The options are: **Right**, **B Lower** and **Watermark**.

Display full name

Displays the full name instead of the mask. Ex: day month year.

Use Time picker

It uses the JQuery plug-in to choose the time.

Database Value

Allows to define a value to be saved to the database.

 Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Date and Hour Field

General Settings

Date and Time field configuration Interface.

Data Type

Define the type of field. When setting it to Date and Time, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server Date and Time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Browser autocomplete

Specifies whether the field will allow the browser’s autocomplete feature. By default, autocomplete is enabled for fields.

- When **disabled**, the field will allow the browser’s autocomplete.
- When **enabled**, the field’s autocomplete will be blocked.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Values Format

Regional Settings

Allows you to apply regional date formatting settings to the field. When not selected, the Date Separator and Time Separator attributes will be displayed.

Display

Allows you to select the display format of the date/time field.

Minimum Date

Minimum value accepted as input for the field.

- **Minimum Date** :
  - **Fixed Date** : Enter the minimum date that scriptcase will criticize in the date type field in the format as presented.
  - **Current Date** : When clicking on the icon next to the box, scriptcase provides the following options:
  - **Simple Current Date** : Will leave the current date as the maximum date, that is, no one born after the current day will enter the form.
  - **Current Date with Increment** : The minimum date will be the current date (+) the days or months or years that you want to increment.
  - **Current Date with Decrement** : The minimum date will be the current date (-) the days or months or years that you want to decrement.

Maximum Date

Maximum value accepted as input for the field.

- **Maximum Date** :
  - **Fixed Date** : Enter the maximum date manually that scriptcase will criticize in the date type field in the format as it is presented.
  - **Current Date** : When clicking the icon next to the box, scriptcase provides the following options:
  - **Simple Current Date** : Will leave the current date as the maximum date, that is, no one born after the current day will enter the form.
  - **Current Date with Increment** : The maximum date will be the current date (+) the days or months or years that you want to increment.
  - **Current Date with Decrement** : The maximum date will be the current date (-) the days or months or years that you want to decrement.
  - **Display Format** : Enables display of the mode

Minimum time

Maximum value accepted by the field.

- **Fixed Time**: Manually enter the maximum time that scriptcase will input in the time type field in the format as it is presented.
  - **Current Time**: When you click the icon next to the box, scriptcase provides the following options:
  - **Current Time with Increment**: The maximum time will be the current time (+) the days or months or years that you increment.
  - **Current Time with Decrement**: The maximum time will be the current time (-) the days or months or years that you want to decrement.

Maximum time

Maximum value accepted by the field.

- **Fixed Time**: Manually enter the maximum time that scriptcase will input in the time type field in the format as it is presented.
  - **Current Time**: When you click the icon next to the box, scriptcase provides the following options:
  - **Current Time with Increment**: The maximum time will be the current time (+) the days or months or years that you increment.
  - **Current Time with Decrement**: The maximum time will be the current time (-) the days or months or years that you want to decrement.

Hour value

Sets the value to be assigned to the time when it is empty and a date is selected in the date picker.

- **Available values are**: Start of day, End of day and Current time.

Apply date limits to the calendar

Apply Minimum and Maximum Date limits when selecting calendar dates, avoiding the selection of dates outside these limits.

Display Format

Display the field format in the data cell.

Date display position

Available when activating the **Display Format** flag, defines where the fill mode will be displayed. The options are: **Right**, **Below** and **Watermark**.

Display full name

Displays the full name instead of the mask. Ex: day month year.

Group Date and Time

Group date and time in the same field.

Display Calendar

Allows you to display a calendar icon next to the field. This allows you to select the month and year, passing the value defined in the calendar to the field.

New Calendar

Show a new calendar with jquery, or the old calendar.

Years Limit

Number of years that will be displayed on the calendar.

View week number

Display week number in application.

Additional months

Display additional months in the calendar.

Available options are: 1, 2 and 3.

Show Combo year and month

Show Combo Year and month in calendar

Database Value

Allows to define a value to be saved to the database.

 Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of

Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - HTML Image Field

General Settings

HTML Image field Configuration Interface.

Data Type

Define the type of field. When setting it to HTML Image, it allows to display an image into the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Image

Set it to display an image. The icon “Select Image” lists all images from Scriptcase and your uploaded images. The “Upload an image” option allows you to send a copy to the Scriptcase server.

Border

Define the width of the Image border in Pixels.

Width

Define the image width size in Pixels.

Height

Define the image height size in Pixels.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.

- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is

being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Label Field

General Settings

Label field Configuration Interface.

Data Type

Define the type of field. When setting it to label, it displays the value with no input to change it.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Text

You can inform a text to be displayed beside the field.

Reload

It links the informed text to reload the form when clicking on it.

Position

It allows positioning the field in the block.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.

- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Card Field

General Settings

Credit Card Number Configuration Interface.

- **Data Type** : Defines the type of field for the application. When it is defined as a Credit Card Number, the field verifies if the value is valid.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Creates a placeholder on the field with the text informed.
- **Initial Value (type)** : Set the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Defines the amount of characters allowed for the field.
- **Validation Image** : Display an image next to the field when it is being validated.
- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to determine the max amount of characters to be typed in.
- **Field Mask** : Defines the display mask for the field. There are three types of masks that can be merged.

Character	Description
9	Represents a numeric character (0-9)
A	Represents an alpha numeric character (A-Z,a-z)
*	Represents any alpha numeric character (A-Z,a-z,0-9) typed by the user.

It is possible to merge two or more masks simultaneously, separated by a semi coma with the smallest mask at start. The replacement occurs when the user is typing when the lowest amount of character exceeds.

Examples of Masks

Field	Mask	Informed Value	Formatted Value
Telephone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Software Key (Only Numbers)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters an Numbers)	**---**	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plates	AAA - 9999	QWE1234	QWE - 1234
ScriptCase License	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple Masks (Telephone)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

- **Hidden Field** : This option makes the field hidden, but still allows it is value to be processed through JavaScript or PHP.
- **Label Field** : This option make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Display the data type of field in the database.

Database Value

Allows to define a value to be saved to the database.

 Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.

- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Card Type Field

General Settings

Credit Card field configuration Interface.

Data Type

Define the field type to Credit Card. It sets a combo-box for selecting the credit card type.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Field size in database

It determines the field size in bytes. It limits the max quantity of the allowed characters.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

 Label Configuration Interface.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn't chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn't chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn't chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Database Value

Allows to define a value to be saved to the database.

 Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookups Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

```
SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
```

The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position

(left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - E-mail Field

General Settings

Email field configuration Interface.

- **Data Type** : Defines the type of field for the application. When set to Email, the field applies validations for an email format.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Defines the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.

- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Defines the amount of characters allowed for the field.
- **Validation Image** : Displays an image next to the field when it is being validated.
- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Defines the name for the session variable that will receive the field value.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through Javascript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Display Icon** : Displays the Email icon next to the field.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - URL Field

General Settings

URL field configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Initial Value (type)

It allows you to define the initial value for the field when in insert mode. The options are:

- **Defined Value**: The field receives the value informed into the input option.

Initial Value Configuration Interface.

- **System Date**: The field receives the current server time.

Amount of characters

Define the number of characters allowed for the field.

Validation Image

Allows displaying an image next to the field when validating it.

Save Variable

It allows saving the value of the field in a session variable to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the session variable that will receive the field value.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Label Field

It sets the field like a Read-Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

Display Icon

Displays an icon beside the field.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

Show the URL as a clickable link

It sets the content of the field as a clickable link.

Target handling where the link will open

Target to open the link when clicking on it.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only

when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**

- **Font** : Allows to choose the font type, that will be applied to the application field.
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - HTML Color Field

General Settings

HTML Color field configuration Interface.

- **Data Type** : Select the type of field for the application. When set to HTML Color, you can select a color to be used in the form.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Set the initial value to the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Defines the amount of characters allowed for the field.
- **Validation Image** : Displays an image next to the field when it is being validated.
- **Save Variable** : Save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Defines the name for the session variable that will receive the field value.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of the field in the database.

Values Format

Format of Values Interface.

Credit Card

It allows to define which cards will be listed and their internal value.

Use Title/Deny

It allows informing a new value if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen. It defines the label and value with the Negative Value and the Title.

Negative value

The value of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Title

Label of the new option if the previous options (American Express, Diners, MasterCard, and Visa) wasn’t chosen.

Related Field

It allows informing which fields contain the values of Credit Card Number so that it can do the validations correctly.

Reload

It Reloads the form when modifying the value.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Lookup Settings

Method used to place a description next to the field.

- **Use lookup to display the field description.** : When enabled, the lookup settings is enabled and opens more settings.
- **SQL Command** : Defines the SQL command that is going to recover the vales from the database. To build a SQL command, it is possible to use the SQL Builder tool. The SQL format needs to have the following format:

  ```
  SELECT Field_displayed FROM table WHERE Key_Field = '{Form_Field}'
  ```

  The Form field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Font** : Defines the font used to display the lookup.
- **Font Color** : Defines the color in HEX for the lookup display.
- **Font Size** : Defines the font’s size in the lookup display.
- **EOF Message** : Defines a message that will be displayed if there is not any results for the record.
- **Use in validation** : When enabled, it will generate an automatic validation based on the lookup, in case there is not any records found, the message will display as a warning (when Updating or Including a record).

- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - HTML Editor Field

General Settings

HTML Editor field configuration Interface.

- **Data Type** : Defines the type of field for the application. When set to HTML Editor, you can inform any type of character and they will be saved in HTML form.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Height** : Set the Height in pixels to the HTML Editor field.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through Javascript or PHP.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **SQL Type** : Displays the data type of field in the database.

Toolbar

HTML Editor toolbar settings Interface.

- **Properties**
  - **Position** : Location of the HTML Editor toolbar.
  - **Alignment** : Toolbar button alignment.
  - **Status** : Status Bar Display (Do not Display, Top and Bottom).
  - **Amount** : The Amount of lines of the HTML Editor toolbar.
- **Button Organization** : Positions the toolbar buttons.
- **Preview** : You can visualize the toolbar according to the previous settings.

Database Value

Allows to define a value to be saved to the database.

Database Values configuration Interface.

- **Defined Value** : Allows to specify the fixed value associated to global variables.

Defined Value configuration Interface.

- **Auto Increment (automatic)** : Allows to use the increment generated by the database. Used only when the field of the database is the type AUTO INCREMENT or similar. For the databases that use sequence like Oracle, PostGres and Firebird, it is necessary to inform the name of the sequence.

Auto Increment configuration Interface.

- **Auto Increment (manual)** : The application generated simulates an auto increment on the field. To include a value to the field it will automatically calculate the next id.
- **Date of Insertion** : When inserting a record, the field will contain the server date as a value.
- **Data of Update** : When updating a record, the value of the field will be the date of the server.
- **User IP** : The field will receive the IP of the machine that the application is be accessed by.
- **Calculated by the database** : The value of the field will be applied by the database. The field will not be used in update processes. For example: it will be updated by a Trigger.
- **Null** : The field will be informed the value null.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**

- **Font** : Allows to choose the font type, that will be applied to the application field title.
- **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Localization Field

General Settings

Location field Configuration Interface.

Data Type

Define the type of field. When setting it to Location, you can see a Combobox with the Languages that are part of the Project.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Reload

It links the informed text to reload the form when clicking on it.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Theme Field

General Settings

Theme field configuration Interface.

Data Type

Define the field type to Theme. it shows a combobox with the list of themes set in your project.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Reload

It Reloads the form when modifying the value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**

- **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
- **Font Size** : Allows to choose the the font size, that will be applied to the application field.
- **Font Color** :Allows to choose a color for the font from the color pallet.
- **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Youtube Field

General Settings

YouTube field Configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Watermark

Create a placeholder to specify a short hint that describes the expected value of an input field

Amount of characters

Define the number of characters allowed for the field.

Display Mode

Define the mode to display the video in the application.

Width

Video Width in pixels.

Height

Video Height in pixels.

Disabled Field

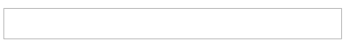
Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.



- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.

- **Border Size** : Applies the border size to the title of the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
- **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Google Maps Field

General Settings

Google Maps field Configuration Interface.

- **Data Type** : Select the type of field for the application. When it is defined as Google Maps, it will use the Google Maps API to display the map in the Form Applications
- **Label** : Set the title to the field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Home** : Defines what type of parameters will be used in the API.

Home Configuration Interface.

- **Display Mode** : Indicates the display mode of the map. It can be opened in a Modal or in a new Window.
- **Width** : Defines the width of the map that is going to be displayed.
- **Height** : Defines the height of the map that is going to be displayed.
- **Zoom** : Defines the initial Zoom (available from the Google API) of the Map location.
- **API Key** : API Key for authorization to use Google Maps in the Application. (Required only for the versions 2 or earlier of the Google API.)

  The API Key is a unique key, composed by a string(text) alphanumeric, which is the license to use the service. When you subscribe to use the service, the key is tied to the domain and the directory of the server. All the pages that use the API needs to be in the same directory that was used for the subscription. In case you have a web server on your local machine, you just need to possess a key for testing, and to do this you only need to set http://localhost in the domain of the subscription.

  To get your API Key access the site by clicking here

- **Link Type** : Defines how the link will be displayed.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.

- **Bold** : Applies the bold style to the font.
- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Image (Database) Field

General Settings

Image (Database) field Configuration Interface.

Data Type

Define the type of field. When setting it to an Image (Database), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Image Border

Width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Progress bar

It displays a progress bar when sending the files to the server.

Upload area

It displays a drag and drop area to upload the file.

Upload area icon

Font-awesome icon for viewing in the file upload zone.

Clickable upload area

Hides the upload button and turns the upload area responsible for the upload function.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Display link only

It allows showing a link to open the image in another window with the original size. (with no re-dimensioning)

Open in Another Window

Allows to open the image in another window.

Extensions and upload size

This setting defines the extensions allowed for uploading and the maximum size of each one.

This setting is important for the security of your project, as it prevents unwanted files from being uploaded.

> **Note** If no value is informed, all extensions will be allowed. The file size will be limited by your PHP configuration.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Multi-upload

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.

- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]
- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.
- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.

- **Underline** : Applies the underline style to the font.
- **Border Size** : Applies the border size to the Input Object.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
- **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the Input Object.
- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field's use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.

- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Image (File Name) Field

General Settings

Data Type

Define the type of field. When setting it to Image (File Name), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Image Border

The width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Progress bar

It displays a progress bar when sending the files to the server.

Upload area

It displays a drag and drop area to upload the file.

Upload area icon

Font-awesome icon for viewing in the file upload zone.

Clickable upload area

Hides the upload button and turns the upload area responsible for the upload function.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API

Keep file after uploading via API

It also saves the file in the default folder for images or documents after uploading to a cloud storage API.

Deletion in the storage API

When deleted in the application, the file will also be removed from the folder in the cloud storage API.

Increment file

It inserts a number increment into the file name if there is already another one with the same name in the upload folder.

Subdirectory for local storage

It sets the Sub-folder name of the stored files. It is relative to the directory of Document upload (see the Settings). It is possible to use global variables or local variables to format the name of the sub-folder.

Create Subfolder

It creates the sub-folder if not already created.

Image Caching

Time in minutes to store the image in the cache.

Hide image name

It displays only the image, without the name.

Files Deletion

It Deletes the files from the directory when deleting a record from the database.

Display link only

It allows showing a link to open the image in another window with the original size. (with no re-dimensioning)

Open in Another Window

Allows to open the image in another window.

File Size

It allows defining the field to store the file size in the database.

Extensions and upload size

This setting defines the extensions allowed for uploading and the maximum size of each one.

This setting is important for the security of your project, as it prevents unwanted files from being uploaded.

> **Note** If no value is informed, all extensions will be allowed. The file size will be limited by your PHP configuration.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Multi-upload

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.
- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]
- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.

- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Document (Database) Field

General Settings

Document (Database) field Configuration Interface.

Data Type

Define the type of field. When setting it to Document (Database), all the document files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Label below field

Defines the message to be displayed below the field.

Icon

It displays an icon beside the field to identify the document type.

Progress bar

It displays a progress bar when sending the files to the server.

Upload area

It displays a drag and drop area to upload the file.

Upload area icon

Font-awesome icon for viewing in the file upload zone.

Clickable upload area

Hides the upload button and turns the upload area responsible for the upload function.

File Name

It allows defining the field to store the file name in the database.

File Size

It allows defining the field to store the file size in the database.

Extensions and upload size

This setting defines the extensions allowed for uploading and the maximum size of each one.

This setting is important for the security of your project, as it prevents unwanted files from being uploaded.

> **Note** If no value is informed, all extensions will be allowed. The file size will be limited by your PHP configuration.

Allow download in read only fields

Allow file download even when field is read-only.

Hidden Field

This option hides the field from the application, but its value is still available for JavaScript or PHP processing.

Disabled Field

Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Multi-upload

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.
- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]

- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.
- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.

- **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Document (File Name) Field

General Settings

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

 Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.
- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]
- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.
- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Multi-upload

This option allows to upload multiple files and store them on a table referencing a record from another table.

To use this option you first need to create a virtual field. _

Multi-Upload Configuration.

Amount of columns

The number of columns to display the files in the Field.

Files sorting

It enables the files sorting after the upload.

Position of the delete

Set the position of the checkbox delete files option.

Position on the upload

Set the position of the checkbox delete files option.

Show status

Display the upload status.

Table

Select the table to stores the file information when uploading.

After selecting the table, you can see a field list of the chosen table to associate the parameters for each field when inserting a new record or updating an existing record.

Multi-Upload fields of the table Configuration.

- **File Name** It gets the name of the loaded file.
- **Upload** It gets the binary value of the file.
- **File Size** Field that will storage the file size in the Database.
- **Auto Increment** Use it only when the field in the database is an auto-increment.
- **Manual Increment** The form is responsible for managing the auto-increment, calculating the value of the field before inserting the record.
- **Foreign Key** It associates the file with a value of a field in the current form.
- **Defined** You can specify a constant value or use a global variable. [global_var]
- **Date of Insertion** It gets the data of the server when inserting the record.
- **Datetime of Insertion** It gets the data and time of the server when inserting the record.
- **User IP** It gets the IP of the computer that is accessing the application.
- **Calculated by the Database** The Database calculates the value of the field.
- **Calculated by the Database if empty** If the value is empty in the application, the Database calculates the value of the field.
- **Null** The field will the null value.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of

Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Barcode Field

General Settings

Configuration Interface of the Barcode Field.

- **Data Type** : DataType of the field for the application.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Disabled Field** : Defines if the field will be disabled in the “Insert Mode” , “Update Mode” or in “Insert and Update Mode”.
- **SQL Type** : Database field type.

Values Format

Configuration Interface of the Barcode Field.

- **Type** : Type of Barcode.
- **Text** : Barcode Text for illustration purposes.
- There are **18 types of barcodes**, that are listed below:

Barcode configuration interface.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Select Field

General Settings

Select field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Select, you can select multiple option from a combo box (Select Field).
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.
- **Use Select2** : Uses the new component for data selection, allowing searches within the select.
- **Display Select2 search area** : Sets whether to display the search field within Select2.

Initial Value Configuration Interface.

- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the field size in bytes. It is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through Javascript or PHP.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

  Selecting the lookup type.

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

The value of the key_field will be stored in the table field.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimitation.

  - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Height** : Defines the height for the select object.
- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.
- **Link** : Allows to create a link to another form allowing to manipulate the list displayed on the select field. After the manipulation, the select object it updated automatically.

◦ **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the select field.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Height** : Defines the height for the select object.
  - **Multiple Values (delimiter)**

    You can store various values for the select field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Height** : Defines the height for the select object.

- **Multiple Values (position)**

  Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

  For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

  The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

  Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

  Setting up Multiple Values (position)

  - **Label** : Text that will be displayed in the item list of the select field.
  - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
  - **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
  - **Start** : Starting position of the string that is going to be stored. The first position is always 1.
  - **Size** : Amount of bytes that is going to occupy in the string.
  - **Height** : Defines the height for the select object.

- **Multiple Values (binary)**

  Allows to store a diversity of information selected on the field in only one value.

  In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

  To store the data, it is done an internal sum of the decimal number. See the examples below.

  Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports

Attribute Value	Lookup Description
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

Editing Lookup Configuration Multiple Values (Binary)

- **Label** : Text that will be displayed in the item list of the select field.
- **Height** : Defines the height for the select object.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

Displaying the original lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Lookup Method - Actual value

This lookup is used to list all the values in the selected field.

This lookup will apply a “distinct” to your SQL query.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field's help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field's help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Double Select Field

General Settings

Double Select field configuration Interface.

- **Data Type** : Select the type of field for the application. When set to Double Select you can have multiple options selected.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Defines the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to set the max amount of characters to be typed in.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through Javascript or PHP.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Displays the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, getting these values from the database.

Lookup Settings Display for the field.
Automatic Lookup Interface..

- **SQL Select Statement** : Defines the SQL command that will get the values displayed on the form field. The command needs to have the following structure:

  ```
  SELECT key_field , display_field FROM table
  ```

  The value of the key_field will be stored in the table field.
- **Height** : Set the height(lines) of the field interface.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter than (;).
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the

users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Checkbox Field

General Settings

CheckBox field configuration Interface.

- **Data Type** : Select the type of field for the application. When set to CheckBox, you can have multiple options selected.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Defines the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Save Variable** : Saves a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to determine the max amount of characters to be typed in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through JavaScript or PHP. .
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or F can be presented like Male or Female or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query). *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the CheckBox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.

    - **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
    - **Reload form when value has changed** : With this option selected, the form will refresh if

there is a change to the selected object in the field.

- **Option check all** : Displays two options on the field to check and uncheck all.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Selects another specific connection existing in the project. The select command will be done on the second connection.

- **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).

  - **Single Value** :

    Setting up Single Value Lookup

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Default** : With this option enabled, the selected item will be inserted to the table when committed.
    - **Columns** : Set the amount of columns, for the list of items.

  - **Multiple Values (delimiter)**

    You can store various values for the checkBox field. The values are separated by the informed delimiter. For example: the combination **Sport**, **Cinema** and **Tourism** selected in form will be stored like **E;C;T** in case the delimiter is ; (semi coma).

    Setting up Multiple Values (delimiter)

    - **Label** : Text that will be displayed in the item list of the checkBox.
    - **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
    - **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
    - **Columns** : Set amount of columns, for the list of items.

  - **Multiple Values (position)**

    Stores a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

    For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

    The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1

Label	Value	Start	Size
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in the database would be stored the following value **MSRD**.

Setting up Multiple Values (position)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Negative** : Value to be saved in the table in case there is not any valued selected on the field.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.
- **Columns** : Set the amount of columns, for the list of items.

- **Multiple Values (binary)**

Stores a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

Setting up Multiple Values (Binary)

- **Label** : Text that will be displayed in the item list of the checkBox.
- **Columns** : Allows you to inform the amount of columns, for the list of items.

- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Saves all the items of the list, to use on other fields, just click on Load lookup definition.
- **Load lookup definitions** : Refreshes the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Option check all** : Displays two options on the field to check and uncheck all.

- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.

- **Border Size** : Applies the border size to the field.
- **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
- **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
- **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
- **Width** : To define a width for the field.
- **Height** : To define a height for the field.

- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Radio Field

General Settings

Radio field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Radio, your allowed to select one of the options listed.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Field size in database** : Determines the size in bytes of the fields. It is used to determine the max amount of characters to be typed in.
- **Hidden Field** : This option makes the field hidden, but still allows it’s value to be processed through Javascript or PHP. .
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **Lookup Methods**

  You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query). *Selecting the lookup type.*

  - **Lookup Method - Automatic**

    Lookup used to list the values that will be displayed on the Checkbox field. These values will be recovered dynamically using the Select command.

    Automatic Lookup Interface.

    - **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

      ```
      SELECT key_field , display_field FROM table
      ```

      The value of the key_field will be stored in the table field.
    - **Columns** : Allows you to inform the amount of columns, for the list of items.

- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  - **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

◦ **Lookup Method - Manual**

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

Manual Lookup Interface.

- **Label** : Text that will be displayed in the item list of the radio.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Default** : With this option enabled, the selected item will be inserted to the table when committed.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on load lookup definition.
- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.
- **Use title** : Allows to display a line of a title on the select object (Title Attribute) associated to a value informed manually (Title internal value).

  - **Title internal value** : Value Saved to the field when the line of the title is selected in the object.
  - **Title** : Text displayed on the title of the Select object.
- **Reload form when value has changed** : With this option selected, the form will refresh if there is a change to the selected object in the field.
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

Displaying the original lookup.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Ajax Settings

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Text Auto-Complete Field

General Settings

Text auto complete field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Text auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal Text for the data.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You’ll inform the Defined Value here.
- **Use Select2**: Uses the new component for data selection, allowing searches within the select.
- **Amount of characters for the Select2** : Sets the number of characters to start the search in Select2.
- **Amount of lines for the Select2** : Sets the maximum number of rows to list the search result in Select2.
- **Width for the Select2** : Sets a width for the area for the Select2.

Initial Value Configuration Interface.

- **Amount of characters** : Define the amount of characters allowed for the field.
- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Complete to the Left** : Allows to define the character that will be used to complete the value to the left that the user typed in to the max size of the field defined in the Field size in database option.
- **Field size in database** : Determines the field size in bytes. it is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Validate on submit** : Validate the field only when the form is submitted.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

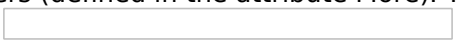
Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed. 

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

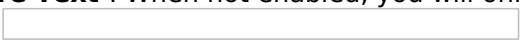
Error message

Message to be displayed if the regular expression is not valid.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).
- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below.  *lookup Settings configuration Interface.*
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  lookup Settings validation configuration Interface.
- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Number Auto-Complete Field

General Settings

Number auto complete field configuration Interface.

- **Data Type** : Define the type of field for the application. When set to Number auto complete, you can inform a value based on the select statement of the lookup settings and it will manipulate an internal number for the data.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Watermark** : Create a placeholder for the selected field.
- **Initial Value (type)** : Allows you to define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
  - **System Date** : The field will receive the current server date.

  If you select the System Date, then it is not necessary to inform the Initial Value attribute.
- **Initial Value** : You will inform the Defined Value here.

Initial Value Configuration Interface.

- **Amount of characters** : Define the amount of characters allowed for the field.
- **Save Variable** : Allows to save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Allows to define the name for the session variable that will receive the field value.
- **Complete to the Left** : Allows to define the character that will be used to complete the value to the left that the user typed in to the max size of the field defined in the Field size in database option.
- **Field size in database** : Determines the field size in bytes. It is used to determine the max size of characters that is allowed to type in.
- **Hidden Field** : This option makes the field hidden, but still allows its value to be processed through JavaScript or PHP.
- **Label Field** : This options make the field behave as a Read Only field, not allowing modifications to its value while in Insert or Update Mode.

Label Configuration Interface.

- **Validate on submit** : Validate the field only when the form is submitted.
- **Disabled Field** : Defines if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : HTML Object that will be used to display the field in the form.
- **SQL Type** : Informs the data type of field in the database.

Values Format

Format of Values Interface

Case Settings

Allows converting the letters of the field when it loses focus. The options are: **Uppercase**, **Lowercase**, **Capitalize the first word**, **Capitalize all words**.

Minimum Length

Determines the minimum number of characters a user must enter in the field.

Integer values greater than or equal to 0 must be provided.

Maximum Length

Determines the maximum number of characters a user can enter in the field.

Integer values greater than or equal to 0 must be provided.

Allowed Characters

Allows to select a set of characters that can be typed into the field. The options are:

- **All** : Allows any type of character to be typed into the field
- **Selected** : Defines a set of characters allowed. Using the configuration below, will be allowed to type letters, numbers and other characters (defined in the attribute More). The letters a, b and c (defined in the Less Attribute) are not allowed.

Regular expression

Regular expression used to validate field content.

Regular expression modifier

Modifiers used in the field content validation regular expression.

Error message

Message to be displayed if the regular expression is not valid.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

Lookup Settings Display for the field.

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field.
- **Rows** : Allows to define the amount of records displayed in the field.
- **Width** : Defines the width the size of capture box (Capture Text).
- **Search options** : Allows to define the search settings of the field(Start equal to, Any part and End equals to).
- **Capture Text** : When not enabled, you will only view the field to inform the data. See an example below. *lookup Settings configuration Interface.*
- **Display original and lookup value** : Displays the value of the field and the value returned from the lookup.

  lookup Settings validation configuration Interface.
- **Show label with the description** : Displays a label with the description.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Ajax Processing

Ajax Processing configuration Interface.

Allows to define the field that will be reloaded when selecting a value that has the (onChange) trigger.

For example: a field of the type select with a **list of states**, and another select field with a **list of cities**. When selecting a state, the cities list is reloaded.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the title of the field.
  - **Height** : To define a height for the title of the field.
- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Signature Field

General Settings

Configuration Interface of the Signature Field.

The signature field will help you creating more sofisticated forms and making it possible to store signatures in your database. Inside our development environment we have specific settings that will help you to customize your field, those options are:

- **Data Type** : You can define the type of field for the application. When it is defined as a text, it accepts letters, numbers and special characters.
- **Label** : Lets you define a label to the field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Background color** : Defines a color to the field background by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Pen color**: Set a color to the pen by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Width** : Lets you define a width to the field.
- **Height** : Set a height to the field.
- **Subtitle** : Defines the subtitle that will be displayed beside the field.
- **Initial Value** : Lets you define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
- **Save Variable** : Save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Disabled Field** : Define if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : Displays the HTML Object that will be used to display the field in the form.
- **SQL Type** : Displays the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of ScriptCase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **CSS of the Title**
  - **Font** : Select the font type, that will be applied to the application field title.
  - **Font Size** : Defines the font size, that will be applied to the application field title.
  - **Font Color** : Choose a color to the font by using a valid hexadecimal color value or from the color picker.
  - **Background Color** : You can define the color for the field by using a valid hexadecimal color value or from the color picker.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Set the underline style to the font.
  - **Border style** : Choose a style for the border.
  - **Collapse** : Defines the collapse for the border.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Choose a color for the border, using a color palette to apply to the title.
  - **Horizontal Alignment** : Position the label of the field in the wanted position (left,right,center and justify).
  - **Vertical Alignment** : Position the label of the field in the wanted position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : Defines a width for the title of the field.
  - **Height** : Set a height for the title of the field.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Rating Field

General Settings

This field allows the developer to create a field for rating using stars (or any other image), where the final user can select a rate option.

Attribute descriptions

Data Type

Define the field type to **Rating**.

Label

The Label option lets you define the title of a field. Example: If the database field name is “**Stars**”, You can display a different name for the user, like “**Stars**”.

Besides use a fixed text, the **Label** attribute allows the use of **Langs** to define the field title, allowing the internationalization of your application.

Label below field

Defines the message to be displayed below the field.

Subtitle

Define the subtitle of the field, below the ratings. **Example**: “Thank you for your feedback!”.

As in the **Label**, the **subtitle** attribute also allows the use of **Langs** for the internationalization of your application.

Amount of icons

It defines how many icons it will display in the field. The value set in this attribute must be according to the evaluation rules.

Initial Value (type)

Allow the definition of an initial value for the field when the form application is in insert mode. The only option available in this field is **Defined**. Selecting this option the attribute **Initial Value** will be showed for set the value.

Example: The initial value is 3 when inserting a new record, the rate field will initiate with 3 stars already selected.

> The value defined in this attribute will overlap any value defined previously.

Save variable

It allows saving the value of the field in a session variable(Global variable) to use it in other applications. For example, you can save the user name in the login form and display its value on the header of other applications.

Variable Name

It allows setting the name of the **session variable** that will receive the field value.

We must inform only the variable name, - **var_rating**.

We can get the value as a global variable.

HTML Type

HTML Object to display the field in the form.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of ScriptCase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **CSS of the Title**
  - **Font** : Select the font type, that will be applied to the application field title.
  - **Font Size** : Defines the font size, that will be applied to the application field title.
  - **Font Color** : Choose a color to the font by using a valid hexadecimal color value or from the color picker.
  - **Background Color** : You can define the color for the field by using a valid hexadecimal color value or from the color picker.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Set the underline style to the font.
  - **Border style** : Choose a style for the border.
  - **Collapse** : Defines the collapse for the border.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Choose a color for the border, using a color palette to apply to the title.
  - **Horizontal Alignment** : Position the label of the field in the wanted position (left,right,center and justify).
  - **Vertical Alignment** : Position the label of the field in the wanted position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : Defines a width for the title of the field.
  - **Height** : Set a height for the title of the field.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Rating Smile Field

General settings

This field type allows the final user evaluate with an emoji that goes from **Very Bad** to **Excellent**.

Data type

Defines the data type of the field in the application. In this case we should select **Rating Smile**.

Label

Defines the title that will be shown in the field when the application is running. The used terms in the interface are essential to the system has good usability, we should use comum name and terms for the final user, instead of the default values of the system.

For example, for the rating smile field, the client would have a better understanding of the field function when the label is **Product evaluation**.

Besides a fixed text, the **Label** allow to use of lang to define the field title, this way is possible to use the application internationalization.

Label below field

Allows you to define a message that will be displayed under the field, which can serve as a help or caption for example.

Subtitle

Define a subtitle that will be shown beside the field. Ex.: Thank you for the feedback!

Below is one example of an application subtitle:

Values and hints

Allow to the developer to define the values to each emoji taht will be saved at the data base and allow to define a hint **(It is show passing the mouse above the emoji)**

Display format

Allows the developer to define the format that the emojis will be shown.

In case the first option is defined, the emojis will be shown like this:

In case the second option is defined, the emojis will be shown like this:

Icon size

Allows defining the icon size in pixels.

Rating padding

Allows for defining the padding value between the emojis.

Icon colors

Allows the icon colors to be defined in each emoji.

Initial value (type)

Allows defining a default value to the field when the application will be in insert mode. Is possible to choose between two options:

Default value: Selecting this option, the default value will be available, where we should input the field's default value. For example, my default value is **Allan**, inserting a new register, the field **Seller Name** will start with Allan.

Save variable

It allows saving a session variable (global variable) with the field value, to be used in other applications.

For example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable name

In this one we should define the variable's name in session, available at the last item, which will receive the field value.

We should inform only the variable's name, **var_seller**.

The recovery of the value is done in the form of a global variable.

Disable field

Allow to Disable field, impossible to change the defined value.

Available options are:
No - Default value, this option don't disable the field.
Update mode - This option disable the field only at update. **Insert mode** -This option disable the field only at insert. **Insert/Update mode** -This option disable the field in both insert and update.

HTML type

HTML object used to show the form field.

SQL type

Infomrs the field type at the data base.

Display Settings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Calendar - Rating Thumbs Field

General settings

This field type allows the final user to save data with the icons like or dislike.

Data type

Defines the data type of the field in the application. In this case we should select **Rating Thumbs**.

Label

Defines the title that will be shown in the field when the application is running. The used terms in the interface are essential to the system has good usability, we should use comum name and terms for the final user, instead of the default values of the system.

For example, for the rating thumbs field, the client would have a better understanding of the field function when the label is **Product evaluation**.

Besides a fixed text, the **Label** allow to use of lang to define the field title, this way is possible to use the application internationalization.

Label below field

Allows you to define a message that will be displayed under the field, which can serve as a help or caption for example.

Subtitle

Define a subtitle that will be shown beside the field. Ex.: Welcome to help us with your feedback!

Below is one example of an application subtitle:

Values and hints

Allow to the developer to define the values to each icon taht will be saved at the database and allow to define a hint **(It is shown passing the mouse above the icon)**

Display format

Allows the developer to define the format that the emojis will be shown.

In case the first option is defined, the emojis will be shown like this:

In case the second option is defined, the emojis will be shown like this:

Icon size

Allows defining the icon size in pixels.

Rating padding

Allows for defining the padding value between the emojis.

Icon colors

Allows the icon colors to be defined in each emoji.

Initial value (type)

Allows defining a default value to the field when the application will be in insert mode. Is possible to choose between two options:

Default value: Selecting this option, the default value will be available, where we should input the field's default value. For example, my default value is **Allan**, inserting a new register, the field **Seller Name** will start with Allan.

Save variable

It allows saving a session variable (global variable) with the field value, to be used in other applications.

For example, in the login form the username can be saved in session and displayed in the header of other applications.

Variable name

In this one we should define the variable's name in session, available at the last item, which will receive the field value.

We should inform only the variable's name, **var_seller**.

The recovery of the value is done in the form of a global variable.

Disable field

Allow to Disable field, impossible to change the defined value.

Available options are:
No - Default value, this option don't disable the field.
Update mode - This option disable the field only at update. **Insert mode** -This option disable the field only at insert. **Insert/Update mode** -This option disable the field in both insert and update.

HTML type

HTML object used to show the form field.

SQL type

Infomrs the field type at the data base.

Display Sttings

The Display Settings allows to define the CSS for the fields individually. For each Display Settings of Scriptcase, exists the same attributes available for this interface.

- **CSS of the Title**
  - **Font** : Allows to choose the font type, that will be applied to the application field title.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field title.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the title.
  - **Horizontal Alignment** : Allows to position the label of the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the label of the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).

- **Width** : To define a width for the title of the field.
- **Height** : To define a height for the title of the field.

- **CSS of the Field**
  - **Font** : Allows to choose the font type, that will be applied to the application field.
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the field, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the field.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the field.
  - **Horizontal Alignment** : Allows to position the field in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the field in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the field.
  - **Height** : To define a height for the field.
- **CSS of the Input Object**
  - **Font** : Allows to choose the font type, that will be applied to the Input Object. For example: Radio, Select, Text, etc
  - **Font Size** : Allows to choose the the font size, that will be applied to the application field.
  - **Font Color** :Allows to choose a color for the font from the color pallet.
  - **Background Color** : Allows to define the color for the Input Object, the color can be selected from the color pallet.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Applies the underline style to the font.
  - **Border Size** : Applies the border size to the Input Object.
  - **Border Color** : Allows to choose a color for the border, using a color pallet to apply to the Input Object.
  - **Horizontal Alignment** : Allows to position the Input Object in the desired position (left,right,center and justify).
  - **Vertical Alignment** : Allows to position the Input Object in the desired position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : To define a width for the Input Object.
  - **Height** : To define a height for the Input Object.

Help Settings

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

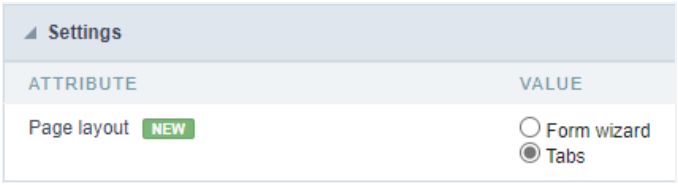
Calendar Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

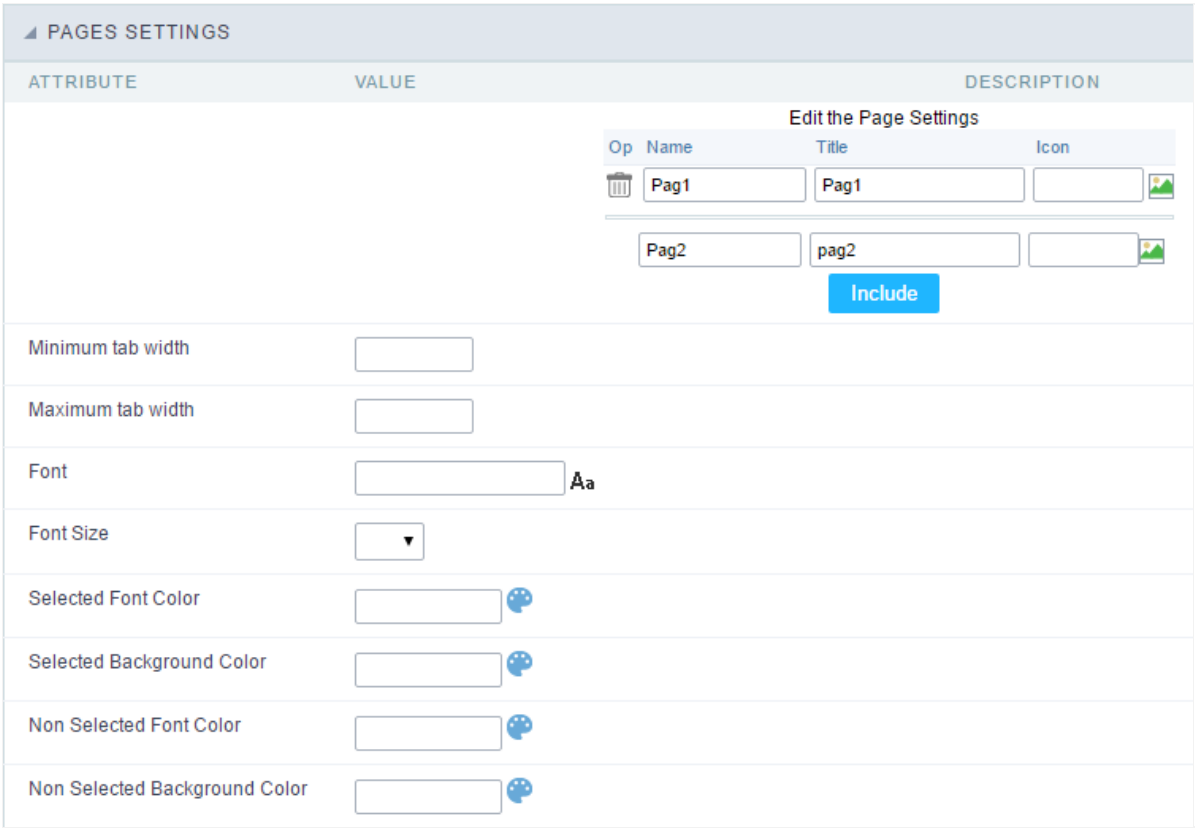
Pages

A page is a container of blocks. Each application has at least one page by default. In Form, Control, and Search applications, there is possible to create many pages. The pages are like Tabs, where each tab contains one or more blocks.

The pages can be shown as **Tabs** or in form of a **Form Wizard**. In the option below you can select the type of your Page Layout.

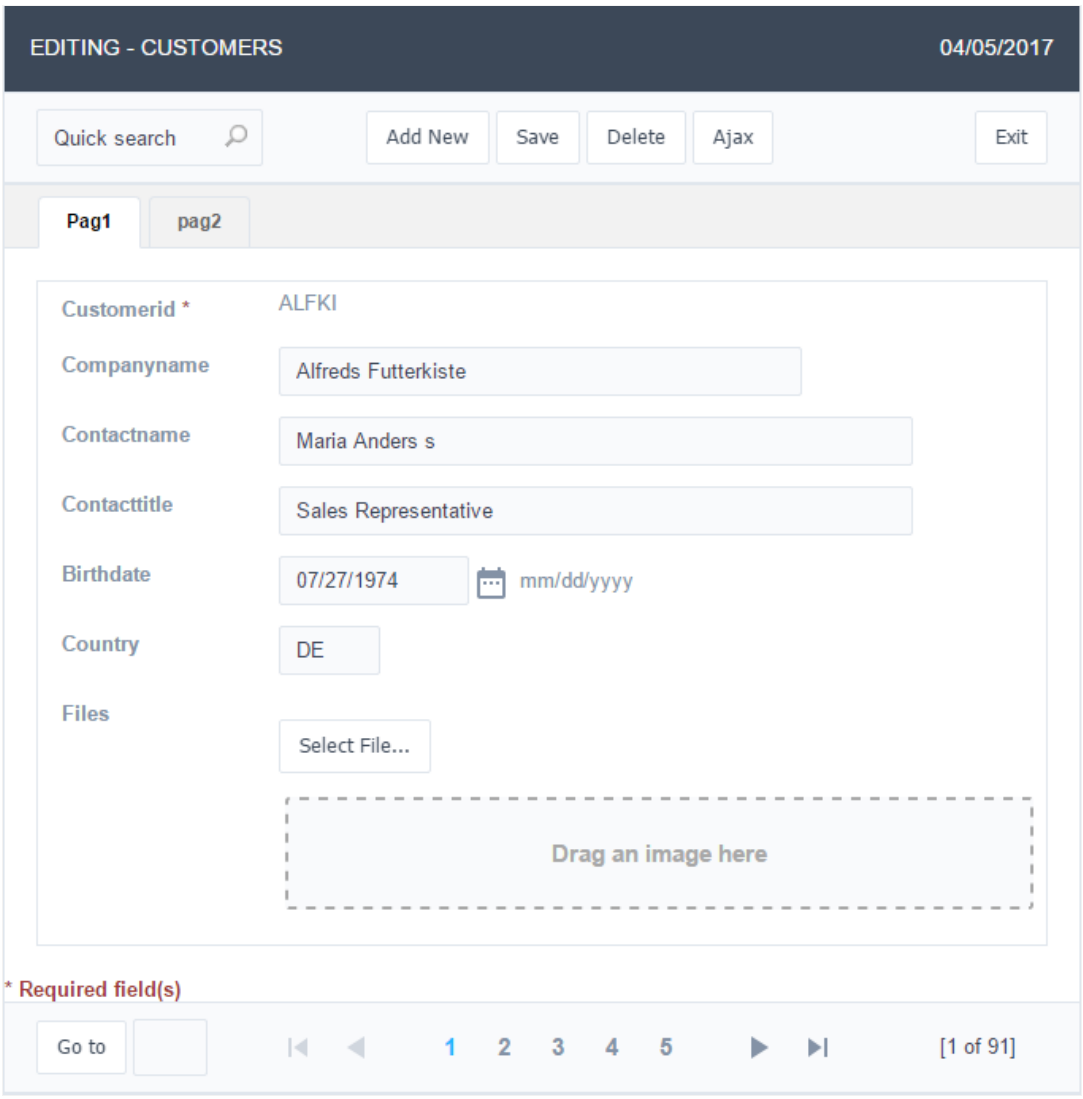


Pages in Tab Format



Pages (available only in the Form, Control and Search applications) configuration Interface.

See the example below of the Form application using two Pages: General Data and Documents.

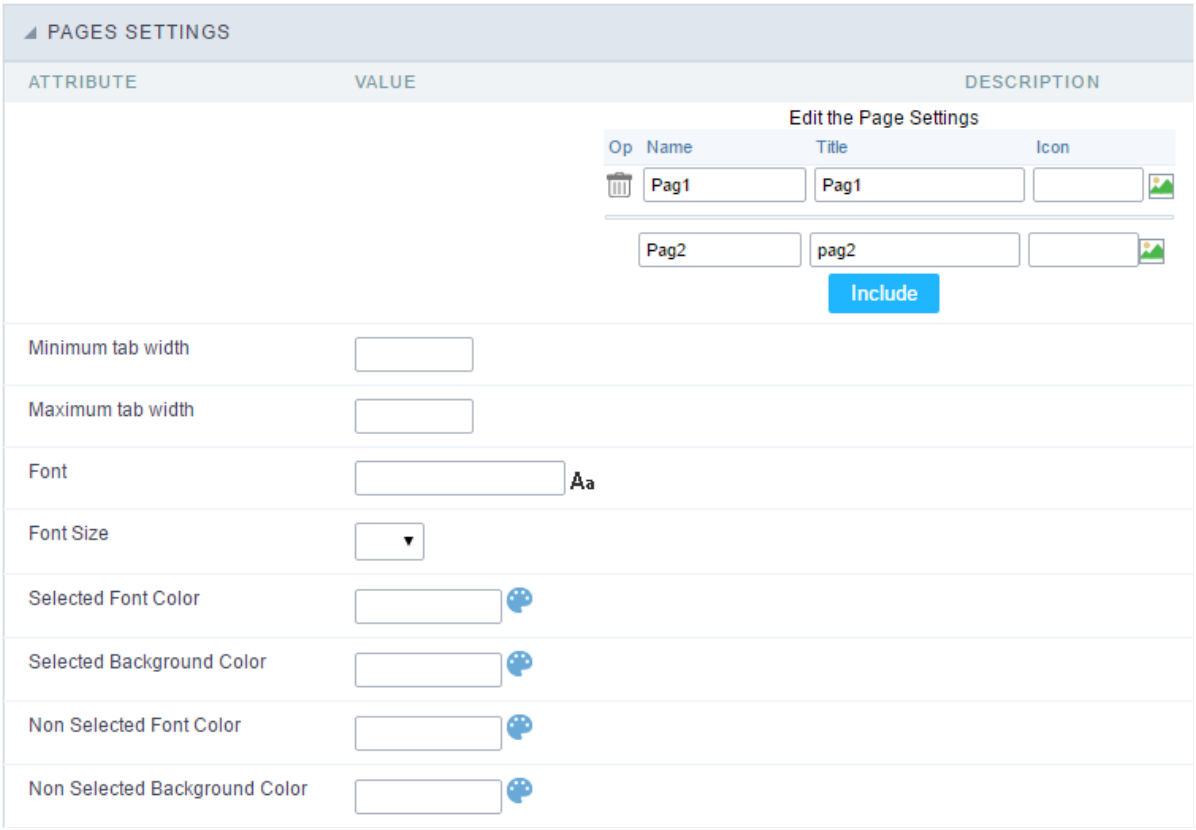


Form Application using Pages

feature.

Pages Settings

The form application already comes with a default page, identified as “Pag1”, but it is possible to rename it. Use the pages when you have an application that contains many fields. A form with more than 20 fields in a vertical way is challenging to use. So you could arrange the fields into the blocks and the blocks into the pages.



Pages Configuration Interface.

Including a new page,

To include a new page, enter the information about the name and the label of the new page, select an image if you wish, then click on the button “Include”.

Deleting a page

To delete a page, click on the Trash icon corresponding to the page line.

Common Settings

Font

Set the font-family of the pages titles. By clicking on the right side icon, you can select the font-family from a list.

Font Size

Set the font size of the pages titles.

Selected Font Color

Set the font color of the selected page.

Selected Background Color

Set the Background Color of the selected page.

Non Selected Font Color

Set the font color of the non selected pages.

Non Selected Background Color

Set the Background Color of the non selected pages.

Blocks

Blocks are “containers” where you can position the application fieldSlides of Forms, Controls, or Grids.

Scriptcase creates applications with one block by default. You can add more blocks as you wish, to organize it in the best way.

See below, the Columns Organization, and where you can define the position of the next block: beside or below the current one.

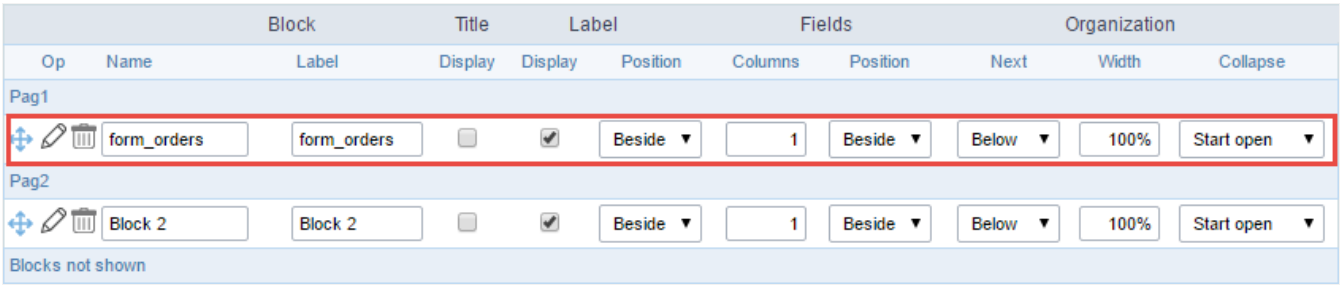
Application Block configuration

On the left side of each block, there are two icons, the first one to edit the information of the block and the second one to delete the block.

Organizing the position of the Blocks

See below how to modify the display order of the Blocks in one Page.

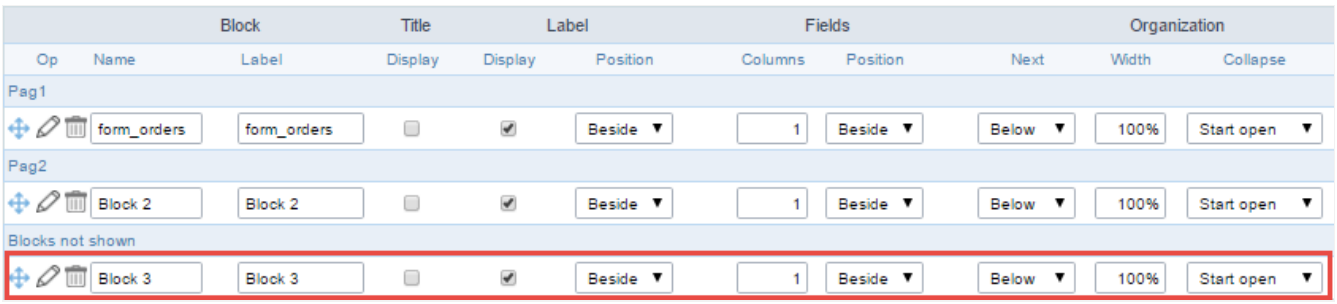
Click and drag the block that you desire to modify to its new position.



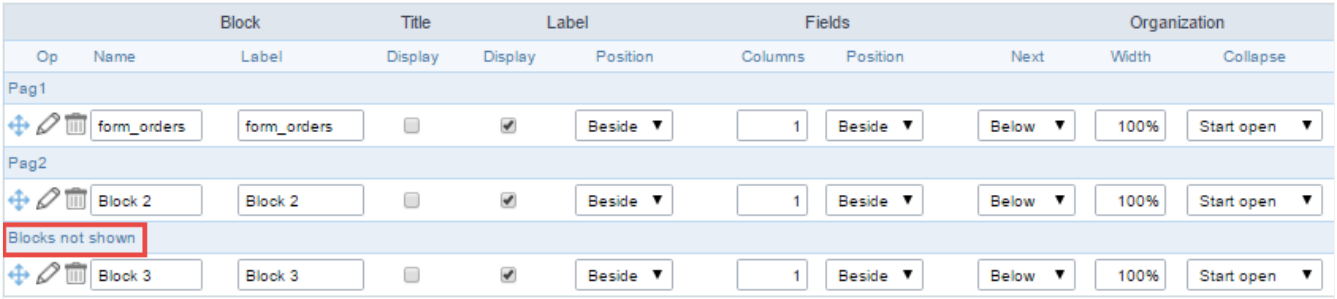
Application Block Display configuration

- See how to remove a block from the display

Click on the block desired and drag it to the item “Blocks not Shown”. This way, you can also drag the block to another page if desired. See the images below.



Application Block Display configuration



Application Block Display configuration

Block

- **Name**: The name that identifier the Block.
- **Label**: Title of the block to display in the application.

Title

- **Display**: It controls the display of the block title.

Label

- **Display**: It controls the display of the field labels of the block.
- **Position**: Options to display label :
  1. **Above**: Display the label above the field.
  2. **Beside**: Display the label beside the field.
  3. **Below**: Display the label below the field.

Fields

- **Columns**: Amount of columns side by side in the block.
- **Position**: The way to display the fields in the block:
  1. **Below**: Display the fields one below the other respecting the number of columns.
  2. **Beside**: Display the fields one beside the other respecting the number of columns.
  3. **Line**: Display the fields one beside the other with no tabulation.

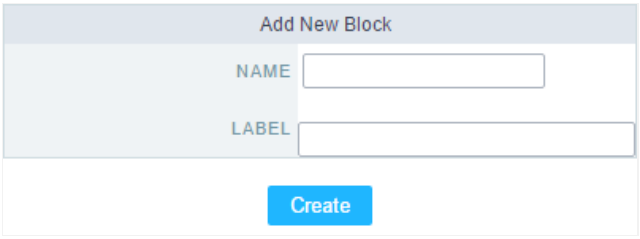
Organization

- **Next**: The way to display the blocks in the page:
  1. **Below**: Set to show the following block below the current one.
  2. **Beside**: Set to show the following block beside the current one.
  3. **Tabs**: Set to show the following block in a different tab then the current one.
- **Width**: Set the block width in pixels or percentages. Use the symbol “%” to indicates the value in percentage.
- **Collapse**: Enables the option to close the block.

Create a New Block

To include new blocks in an Application, click on the button

. Then, enter the name and label of the block in the following interface and finish by click on Create.



Creating application blocks configuration

Name

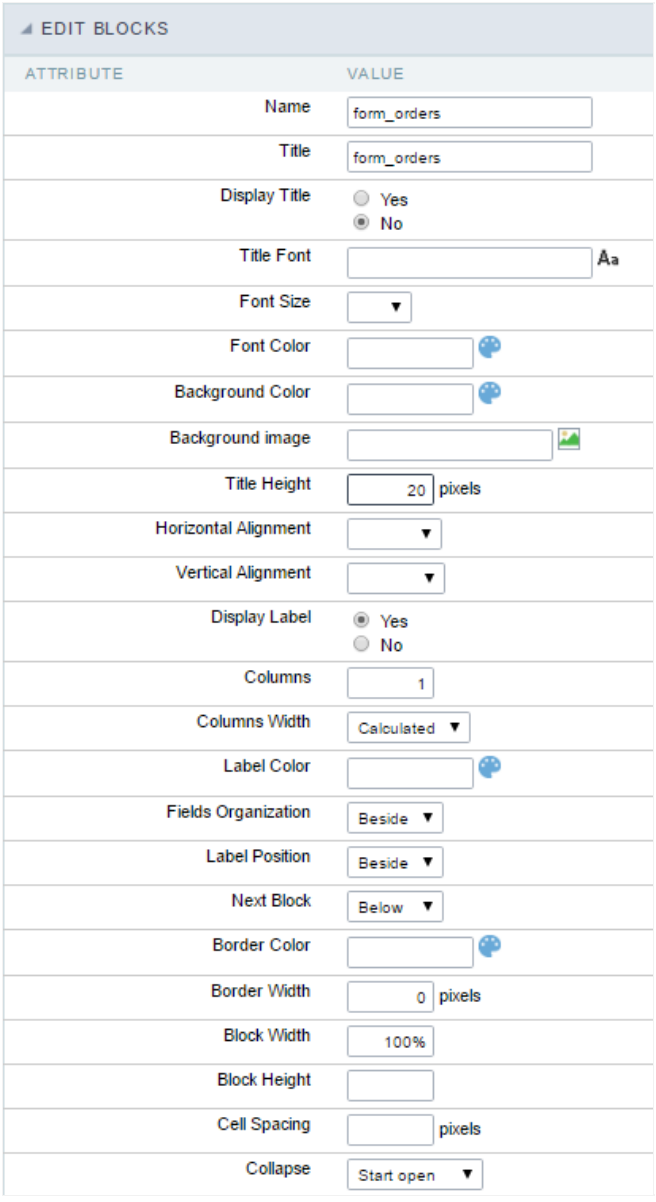
Name of the Block.

Label

Title of the block to display in the application.

Edit Blocks

To edit a block, click on the icon , that is on the left side of the block. Then you can see the following interface to define the parameters of the blocks. Click on Save to finish.



Application Block editing interface

Name

Name of the block. #### Title
Block title for display. #### Display Title
This option, when active, allows displaying the block title. #### Title Font
Set the font family of the block title. #### Font Size
Set the font size of the block title. #### Font Color
Set the font color of the block title. #### Background Color
Set the Background Color of the block title. #### Background image
Set a Background image for the block title. #### Title Height
Height in pixels of the block title line. #### Horizontal Alignment
Horizontal Alignment of the block title (Left, Center, and Right). #### Vertical Alignment
Vertical Alignment of the block title (Top, Middle, and Bottom). #### Display Label
Display the labels of the fields in the block. #### Columns
Amount of field columns in a block. #### Columns Width
Set the field column width of the block. #### Label Color
Color of the field labels. #### Fields Organization
The way to display the fields in the block. #### Label Position
Set the position of the field labels of the block.

The options are:

- **Beside** - This option positions the label on the right side of the field.

- **Above** - This option places the label above the field.

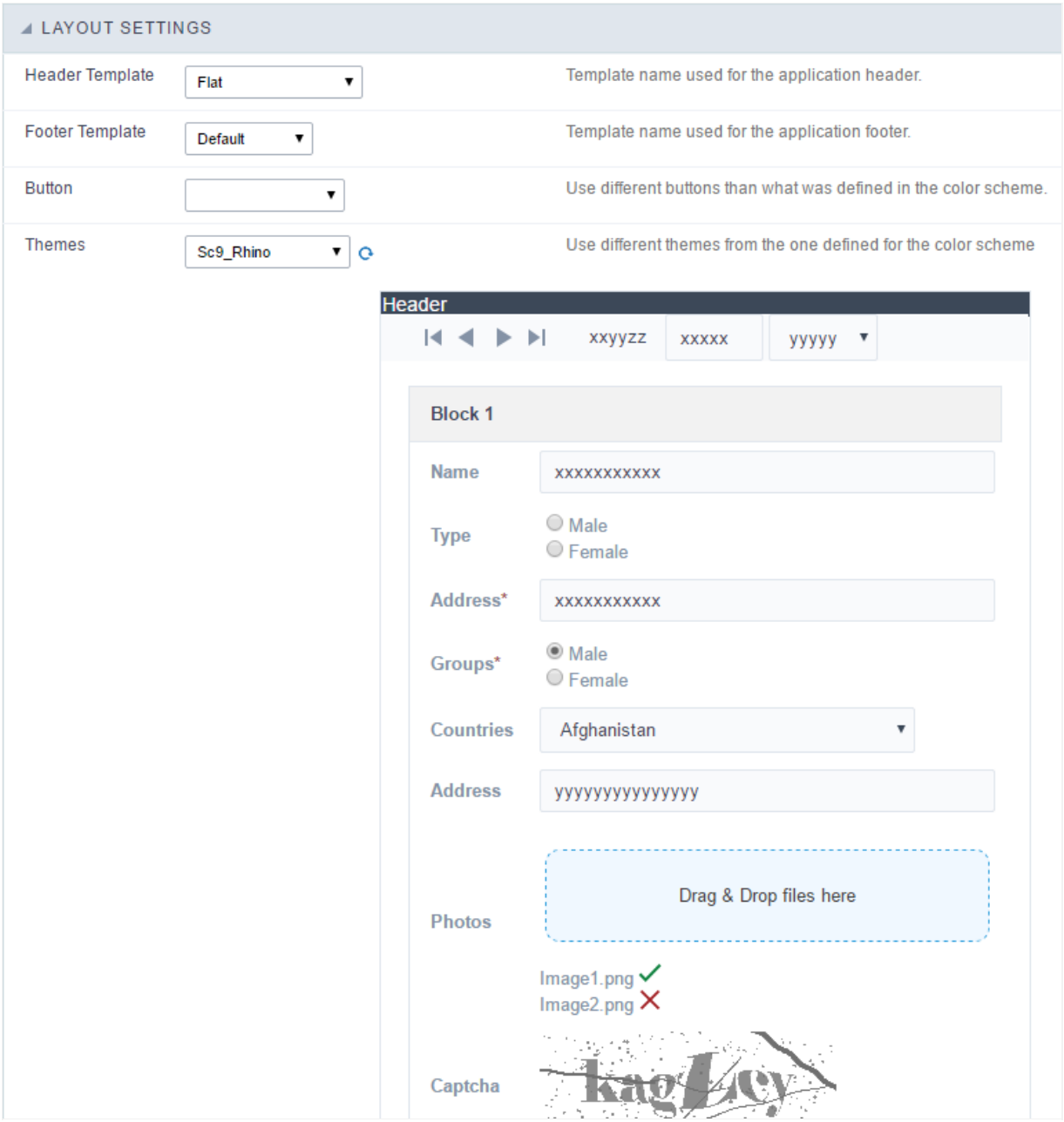
- **Below** - This option places the label below the field.

Next Block

Set the position of the following block relating to the current one. #### Border Color
The border Color for the block. #### Border Width
The border Width for the block. #### Block Width
The width for the block. #### Block Height
The Height for the block. #### Cell Spacing
The Cell Spacing in the block. #### Collapse
It enables the option to close the block.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout Settings

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

Allows choosing the button theme for the application.

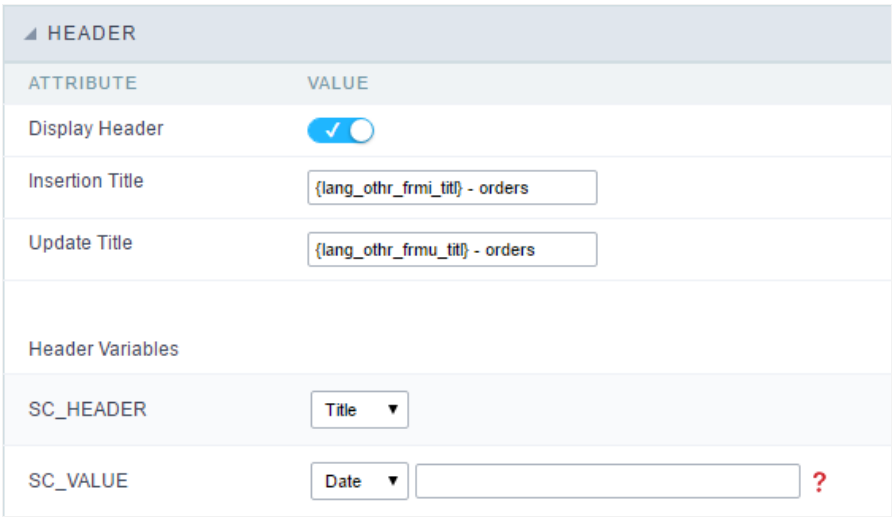
Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks

according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

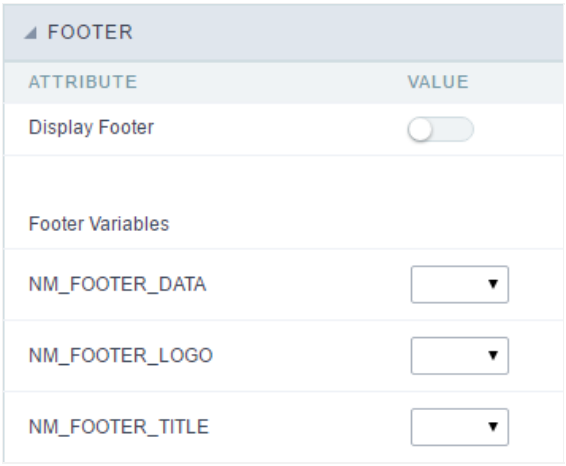
Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon "**Choose Image**", and you still can upload new images by using the button "**Upload**". .
- **Value**: It displays the content of the text input. You can inform static texts and "**Global Variables**". e.g. "Employee Name: [v_name]".

Related Links

Related Videos

Calendar Events Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application's CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event's PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onCalendarApplicationInit - Calendar

This event occurs before the application execute the SQL, and is executed only once.

onApplicationInit - Calendar

This event occurs before the application execute the SQL, and execute only once. It's used to do verification of variables, and security verification. Ex: if ([glo_var_department] != 'financial'){ sc_redir(app_x.php); }

onCalendarScriptInit - Calendar

These formulas are executed before the application starts. At this moment, the application local variables are not available.

onScriptInit - Calendar

These formulas are executed before the application starts. At this moment, the application local variables are not available.

onLoad - Calendar

This event its executed before the form is loaded. In this moment all the applications variables are available.

onRefresh - Calendar

This event occurs when the form application is reloaded, is possible to reload the form based on fields (select , radio, checkbox)

onValidate - Calendar

This event is executed when the form is submitted to the server through the buttons: "include", "change" or "delete"

onValidateFailure - Calendar

This event occurs when submitting the application to onValidate there is an error generated by the macros sc_error_exit, sc_error_message, or by an error verified by Scriptcase (errors related to Database).

onValidateSuccess - Calendar

This event occurs when submitting the application to onValidate, there is no error generated by the macros sc_error_exit, sc_error_message, or by an error verified by Scriptcase (errors related to Database).

onBeforeInsert - Calendar

This event occurs before adding a new row, it is possible to create or call sophisticated data validations on this event.

onAfterInsert - Calendar

This event occurs after clicking the Insert button on the form. The form will execute the onValidate, onValidateSuccess and onBeforeInsert events, and right after executing the SQL command to insert the database record, it will execute the onAfterInsert event.

Example: After the inclusion of a record in the database, we want to store the inserting operation in a log table.

sc_exec_sql(" insert into log values ([glo_user],'target table','insert',{key_field}) "); //insert the user id that was stored in a session variable [glo_user] and the newly added registry key {key_field}.

onBeforeUpdate - Calendar

This event occurs after form validation (onValidate and onValidateSuccess), when clicking the Save button, and before executing the SQL command to Update the record in the database.

Example: We are checking the user privileges table before updating the record, if it does not have such privilege, we send an error message.

```
sc_lookup(priv_upd ," select priv_upd from tb_privileges where login = [var_login] ");

if( {priv_upd [0][0]} != 'YES'){

  sc_error_message("You do not have privileges to perform this operation");

}
```

onAfterUpdate - Calendar

This event occurs after clicking on the Form's Save button. The form will then execute the onValidate, onValidateSuccess and onBeforeUpdate events, and right after executing the SQL command to insert the database record, it will execute the onAfterUpdate event.

Example: After updating a record in the database, we want to store the update operation in a log table.

sc_exec_sql(" insert into log values ([glo_user],'target table','update',{key_field}) "); //insert the user id that was stored in a session variable [glo_user] and the newly added registry key {key_field}.

onBeforeDelete - Calendar

This event occurs before deleting rows on form applications. Ex: Before delete a row it is possible to verify the users privileges, if the user has no delete privilege the transaction is cancelled automatically. sc_lookup(priv_del," select priv_del from tb_privileges where login = [var_login] "); if({priv_del[0][0]} != 'Y') { sc_erro_mensage(" você nao tem privilégios para executar esta operação "); }

onAfterDelete - Calendar

This event occurs after clicking the form's Delete button. The form will then execute the events onValidate, onValidateSuccess and onBeforeDelete, and then, right after executing the SQL command to delete the record in the database, it will execute the event onAfterDelete.

Example: After deleting the record we can easily create logs, according to the code below:

sc_exec_sql(" insert into tb_log values([glo_user],'exclusion','customer_table',{code}); //we are storing the literals 'exclusion' and 'customer_table', the session variable [glo_user] and the field variable {code} containing the deleted content of the code field.

onFooter - Calendar

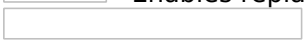
This event is used when in the footer we need to display some calculated value, it is in this event that we write the calculation necessary to assemble some value that we want to display.

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

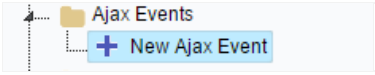
> Click Here to view the Scriptcase hotkeys documentation.

Calendar Ajax events

OnClick

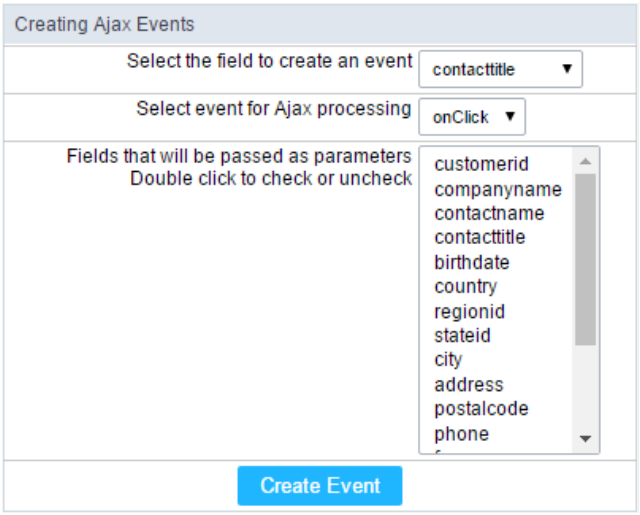
The ajax event OnClick is executed when the field that it's based on is clicked.

- Creating a new ajax event



Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

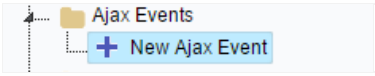
Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

OnChange

The ajax event OnChange is executed when the value of the field that it's based on is modified.

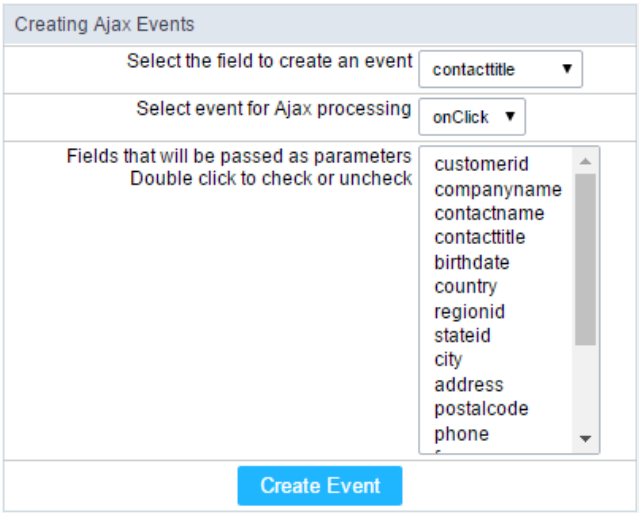
NOTE: The **radio field** is not compatible with this type of event.

- Creating a new ajax event



Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

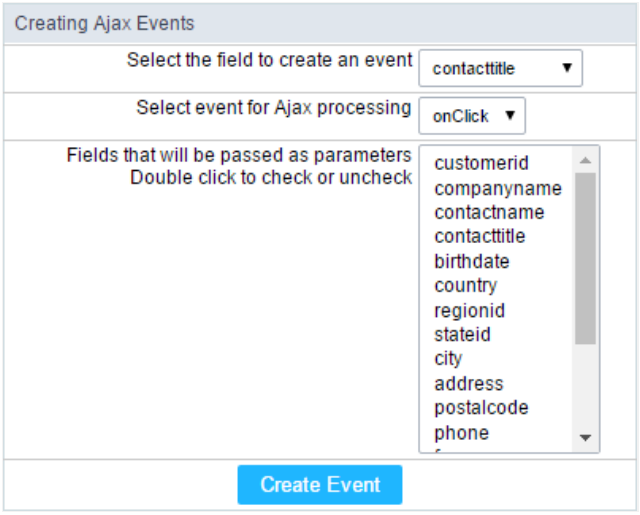
OnBlur

The ajax event OnBlur is executed when the focus is removed from the field that event is based on.

- Creating a new ajax event

Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

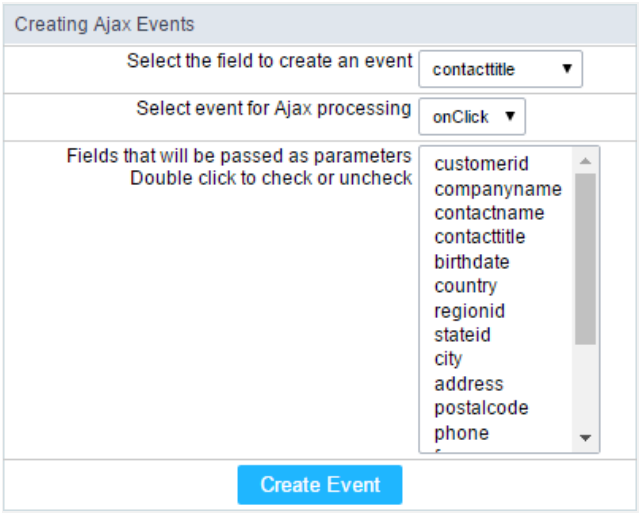
OnFocus

The ajax event OnFocus is executed when the field that it's based on is applied with a focus.

- Creating a new ajax event

Creating a new ajax event

- Selecting a field



Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.

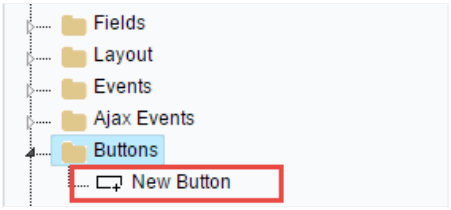
Choose an event that run the ajax Defines which event will be added to the field.

Related Links

Related Videos

Calendar Buttons Settings

Including the default applications buttons, you can also create new buttons manually. These buttons can appear in the application toolbar.

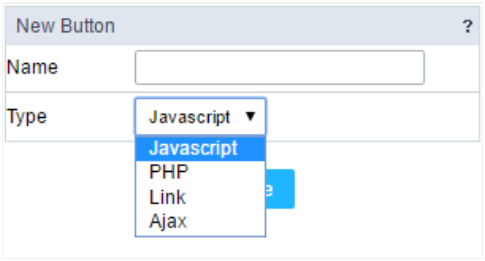


Creating new buttons

Creating a button

To create a new button, click int the option “new button” and type a name and the button type.

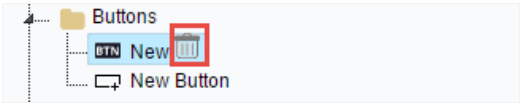
Button types: (JavaScript, PHP, and Ajax).



Button types

Deleting a button

To delete a button click on the icon next to the name of the button in the application menu (recycle bin).



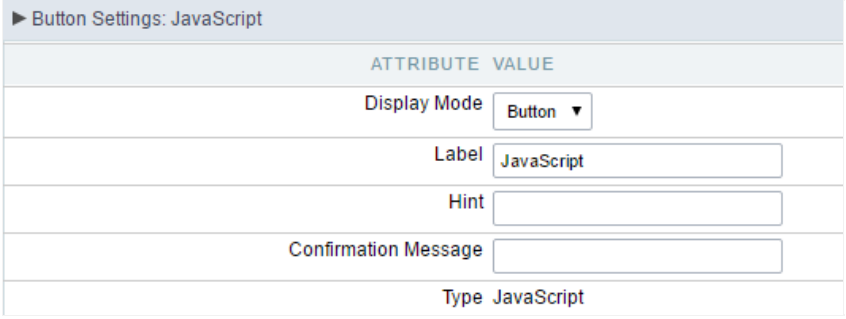
Deleting a button

JavaScript

Display Mode

You can configure the display mode of the javascript button in Image, Button or Link.

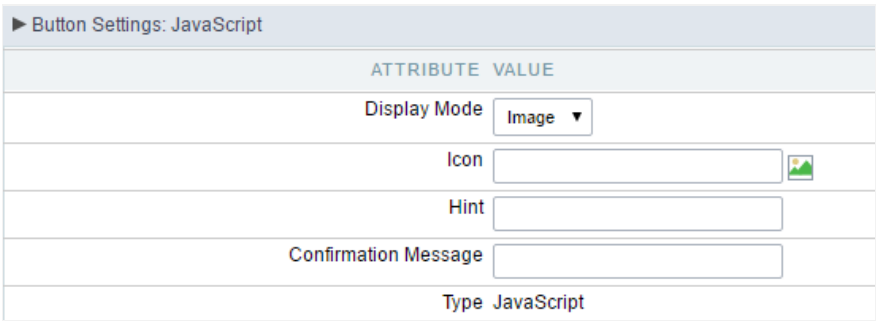
Button



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

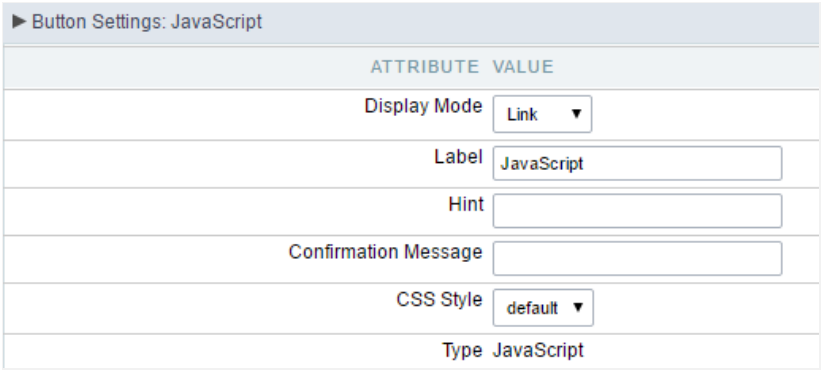
Image



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

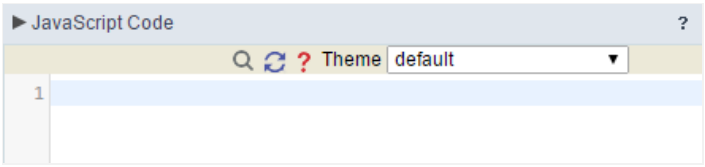
Link



Setting up Javascript Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Code Block



Javascript button coding block.

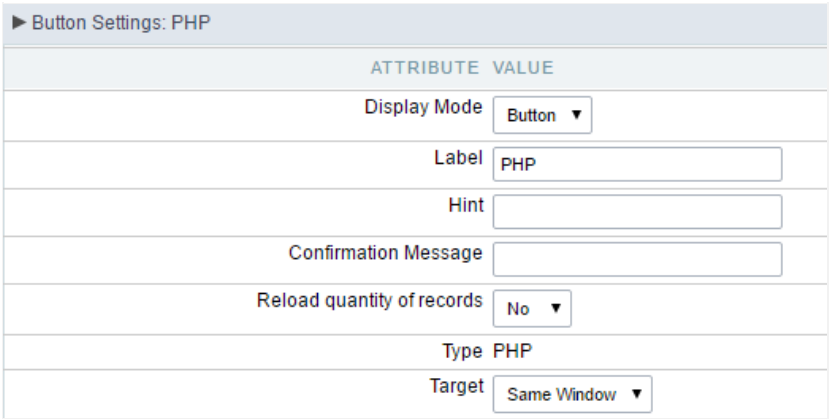
In this block, only JavaScript is accepted.

PHP

Display Mode

You can configure the display mode of the PHP button in Image, Button or Link.

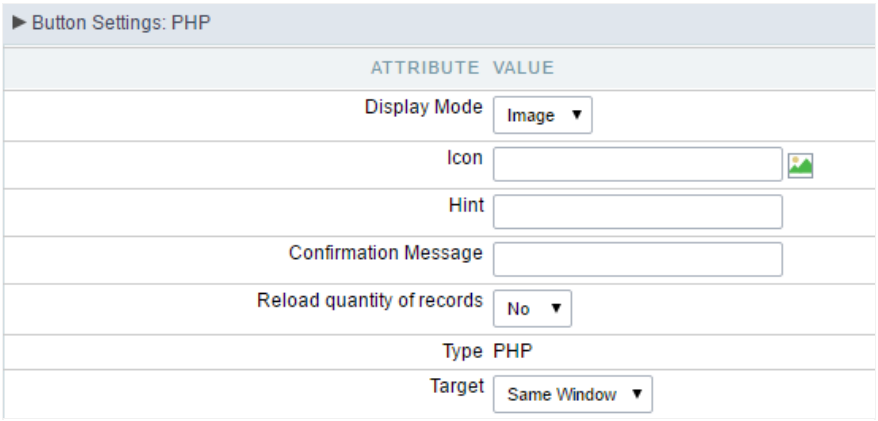
Button



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

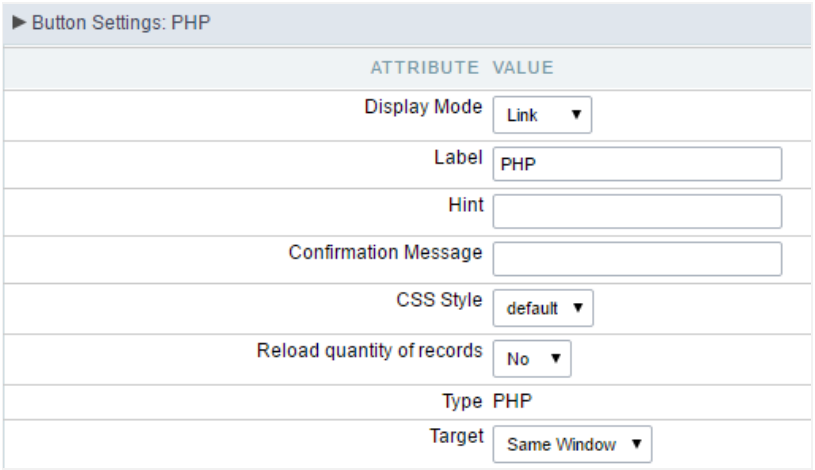
Image



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

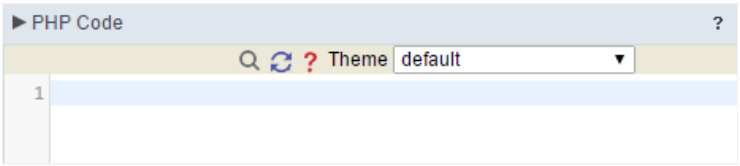
Link



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

Code Block



*Ajax button coding block. *

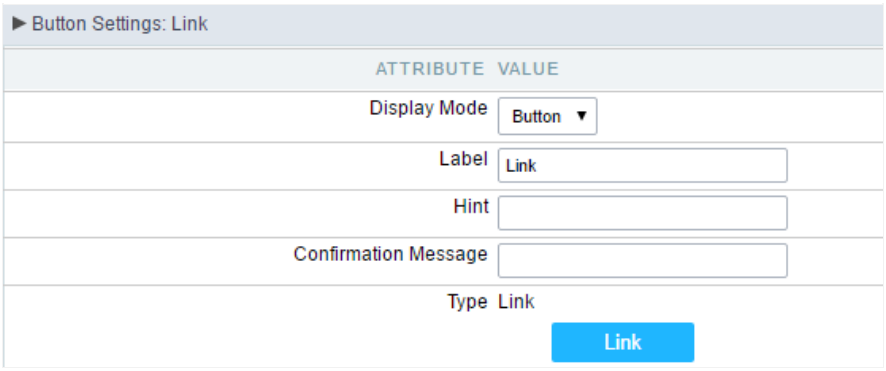
In this block, you can use macros, PHP code and JavaScript.

Link

Display Mode

You can configure the display mode of the link button in Image, Button or Link.

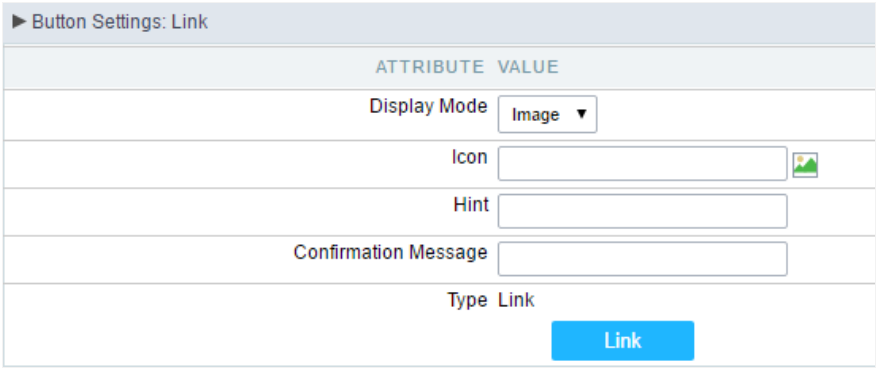
Button



Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Image

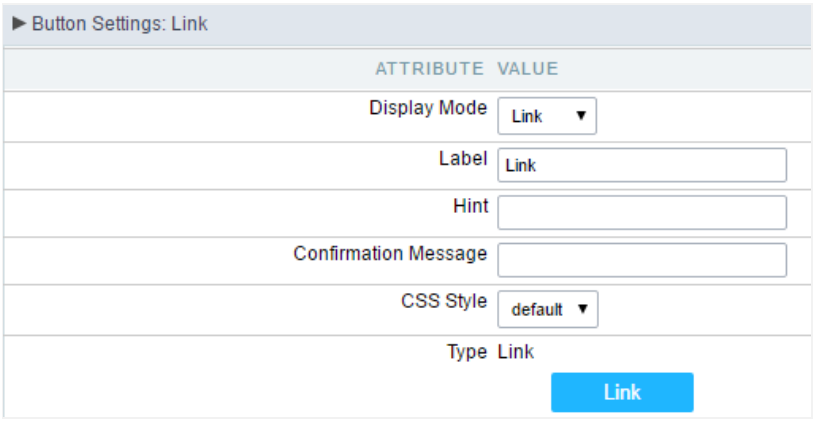


Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.

Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Link

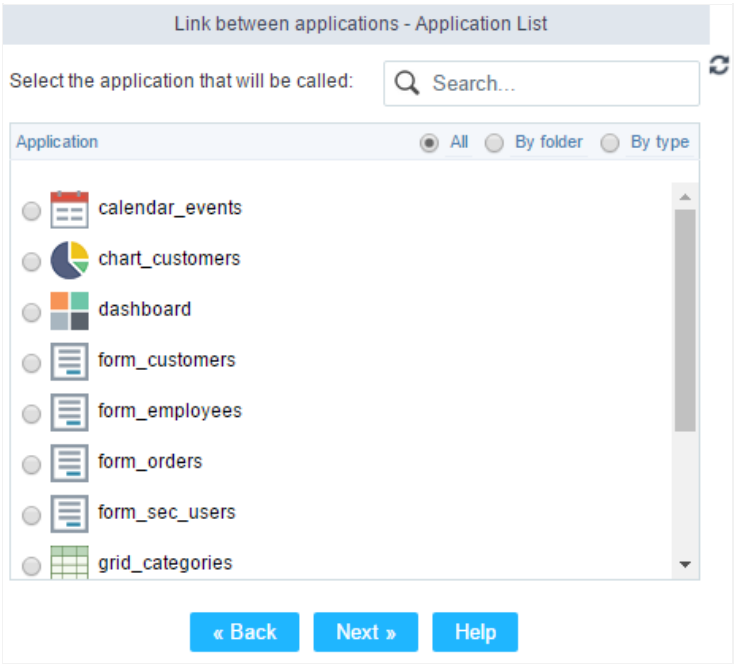


Setting up link Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Setting up the Link

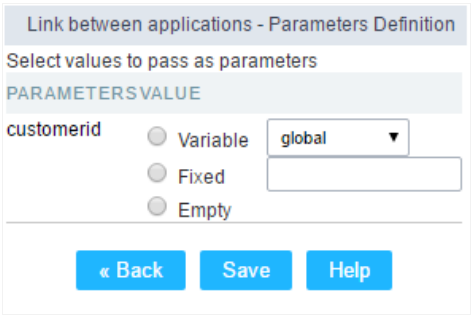
- Selecting the applications



Choosing the application for the button link.

You should select an application to be called from the button link.

- Link Parameters



Choosing the parameters for the button link.

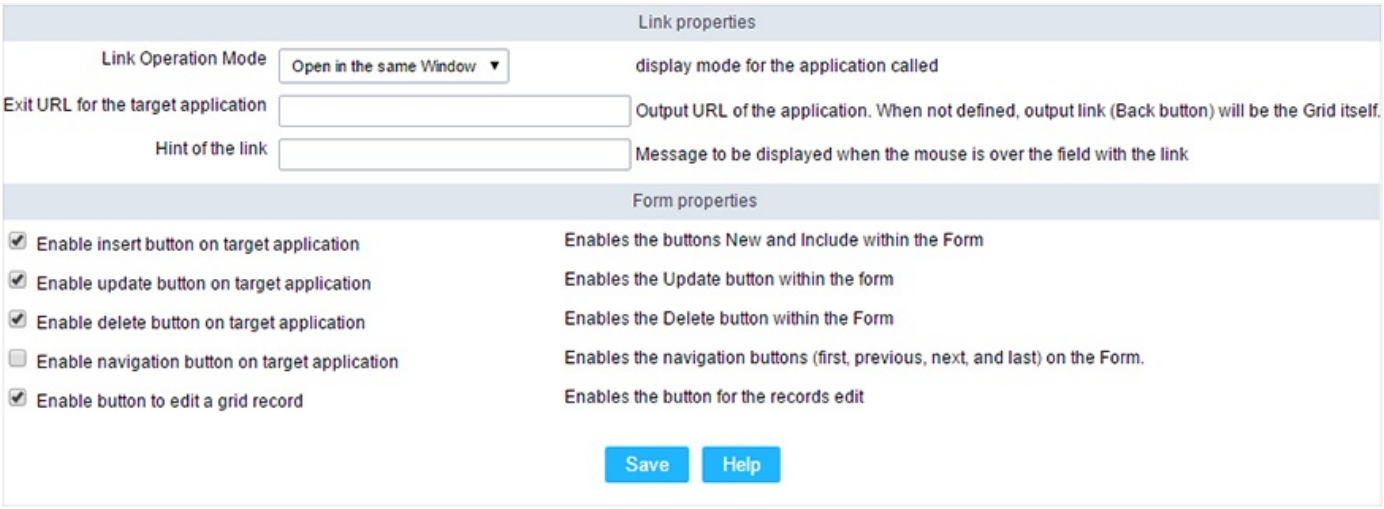
Field Allows you to use an existing field from the current application as a parameter for the link.

Variable Allows you to use a global variable from the current application as a parameter for the link.

Fixed Allows you to inform a fixed value as a parameter for the link.

Empty No value will be passed as a parameter for the link.

- Link Properties (Grid)



Configuring the properties for the link button when the destined application is a Grid.

Link Operation Mode How the link will open (Open in the same Window, Open in another Window, Modal).

Exit URL for the target application URL or an application that be redirect to when exiting the Grid application.

Initial Mode Allows you to define the initial mode of the grid application (Search or Grid).

Number of Lines Allows you to define the amount of lines displayed in the Grid.

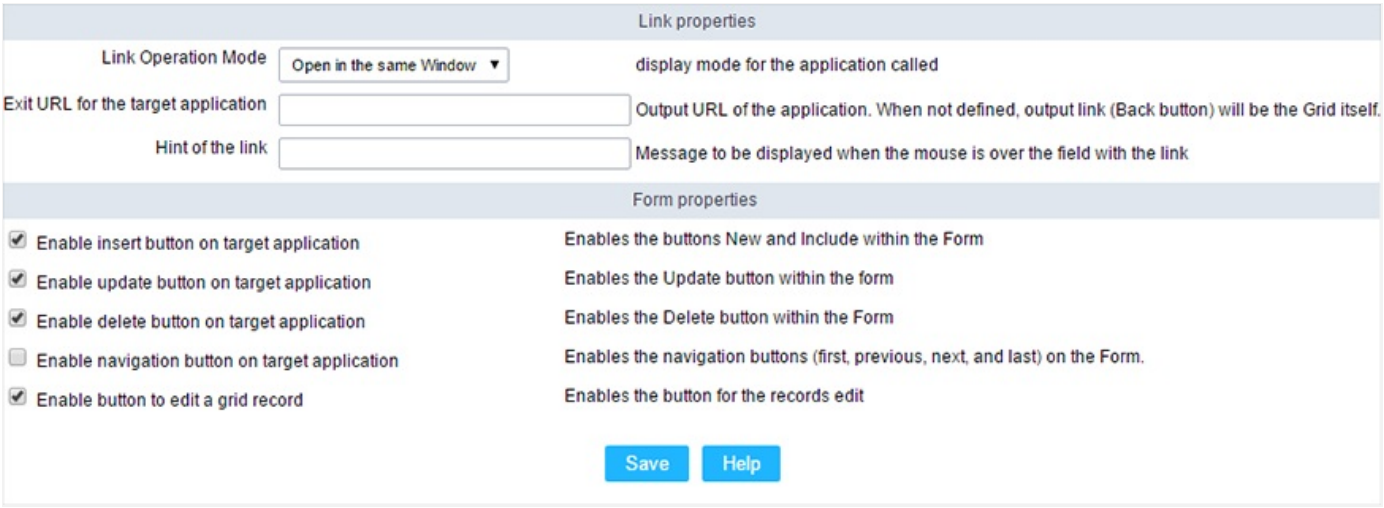
Number of Columns Allows you to define the amount of columns displayed in the Grid.

Paging Enable the paging in the Grid.

Display Header Enable the Grid Header.

Active Navigation Buttons Enable the navigation button (First, Back, Next and Last) in the Grid.

- Link Properties (Form)



Configuring the properties for the link button when the destined application is a Form.

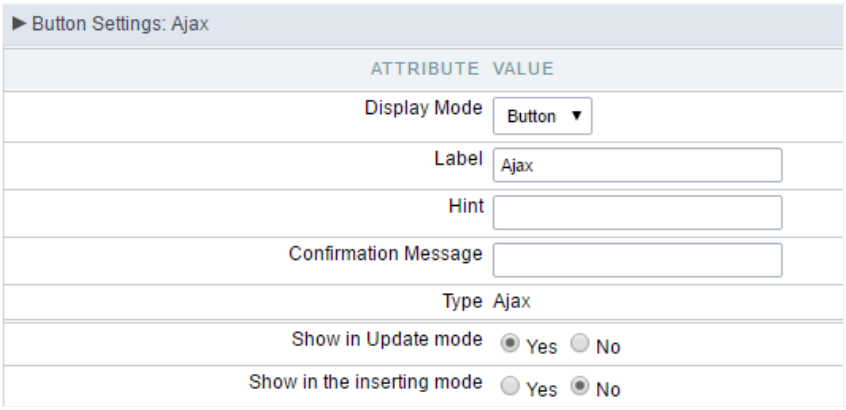
Link Operation Mode	How the link will open (Open in the same Window, Open in another Window, Modal).
Exit URL for the target application	URL or an application that be redirect to when exiting the Form application.
Enable insert button on target application	Enable the “New” button in the Form Application.
Enable update button on target application	Enable the “Update” button in the Form Application.
Enable delete button on target application	Enable the “Delete” button in the Form Application.
Enable navigation button on target application	Enable the navigation button (First, Back, Next and Last) in the Form.
Enable button to edit a grid record	Enable the buttons that allow you to edit the records of a Grid

Ajax

Display Mode

You can configure the display mode of the Ajax button in Image, Button or Link.

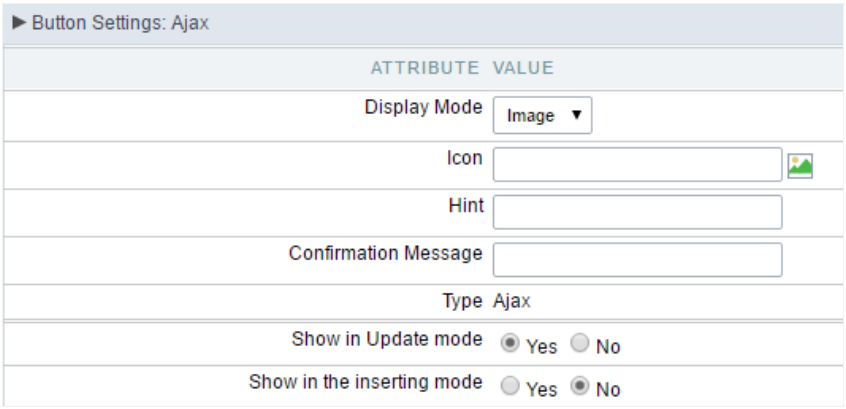
Button



Setting up Ajax Button.

Display Mode	You can select the display mode for the Ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

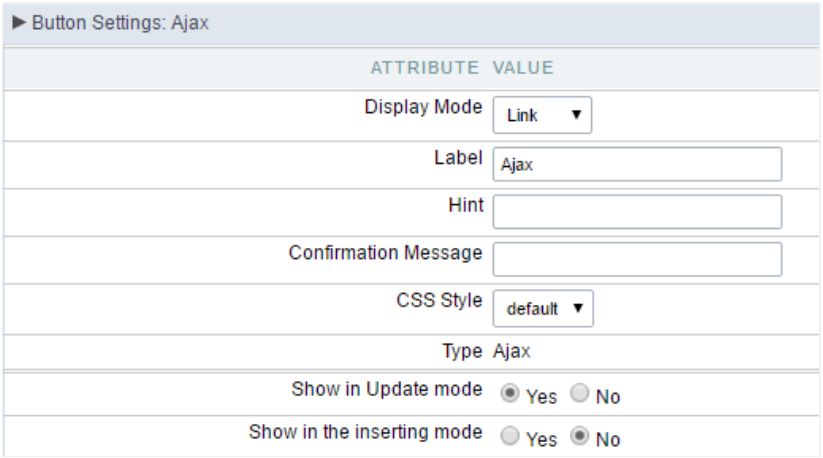
Image



Setting up Ajax Button.

Display Mode	You can select the display mode for the Ajax button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

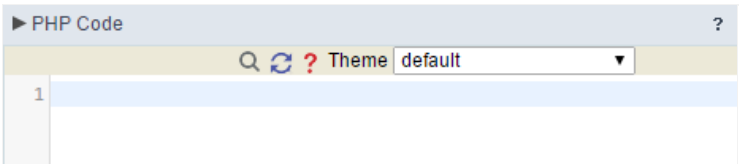
Link



Setting up Ajax Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Code Block



Ajax button coding block.

In this block, you can use macros, Ajax code and JavaScript.

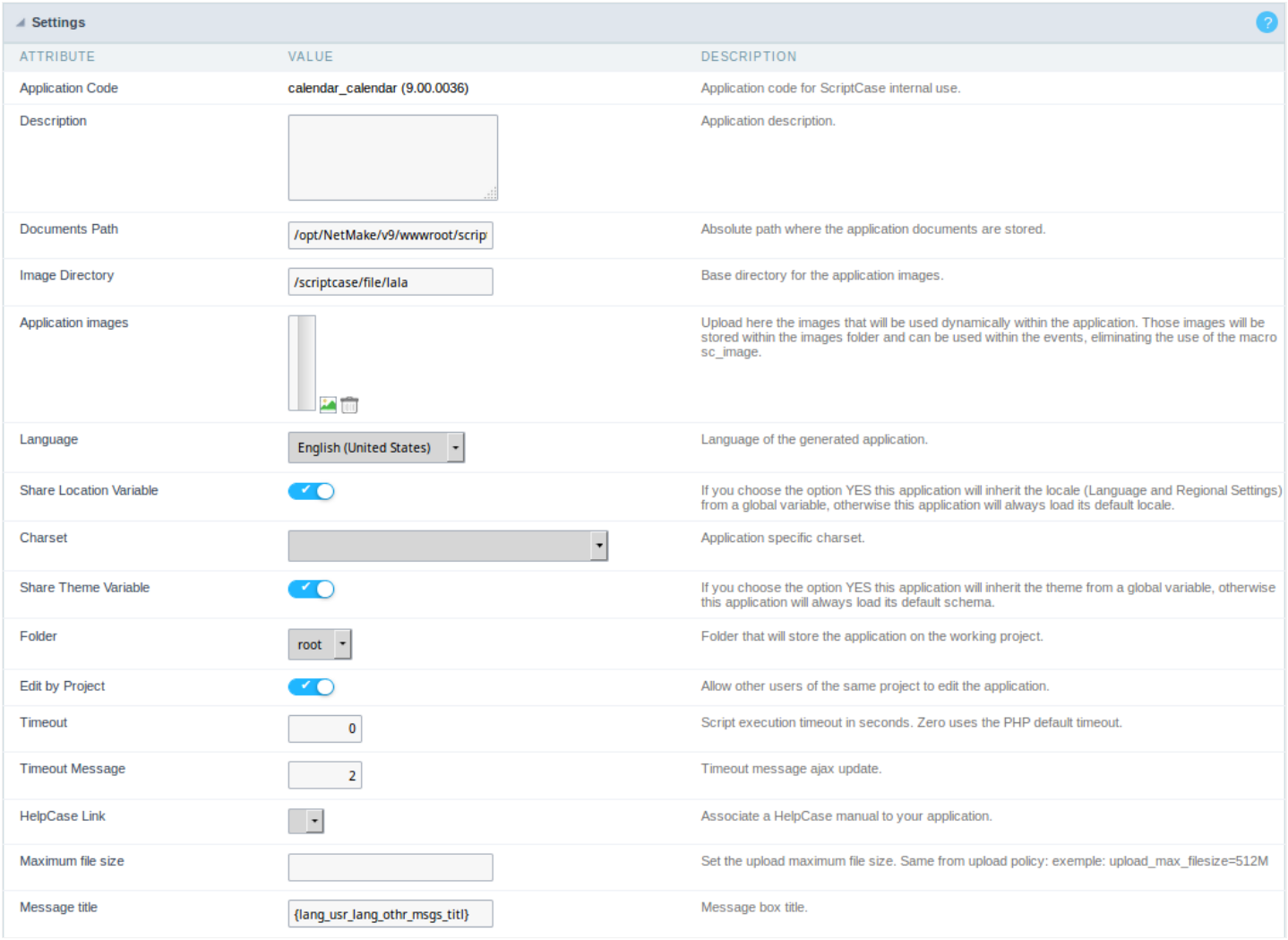
Related Links 🔗

Related Videos ▷

Application Settings (Calendar)

Settings

With this interface, you can set the common attributes of the app.



Application Settings Interface

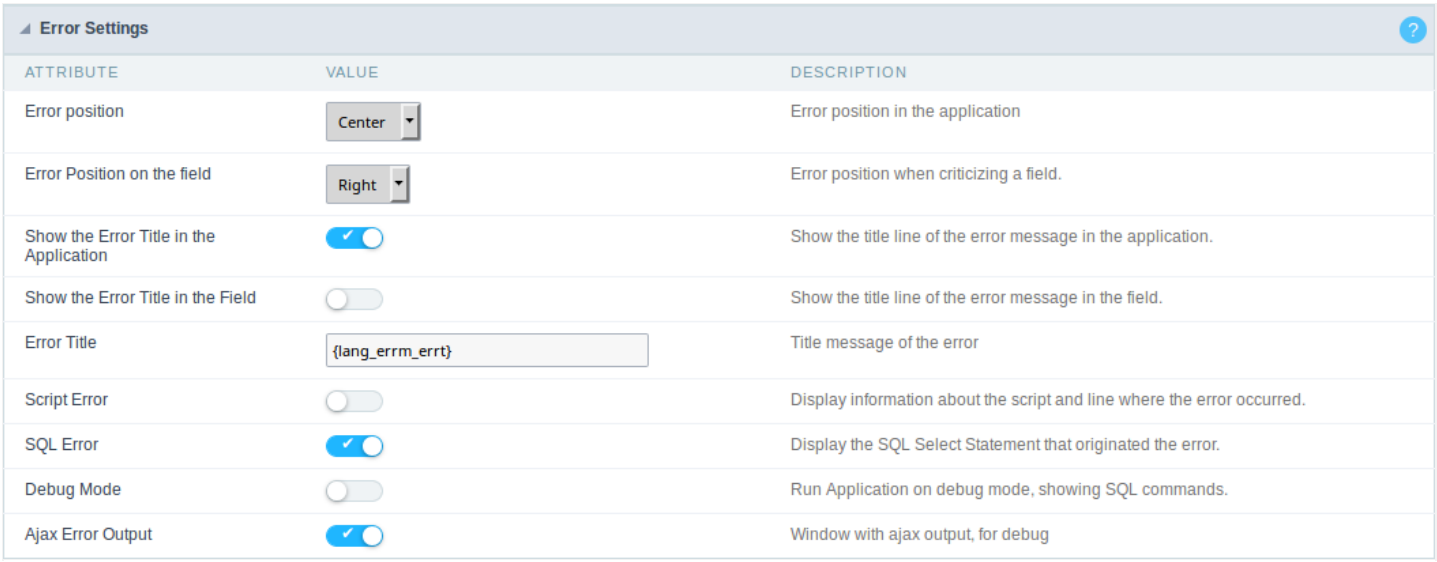
- **Attributes**
  - **Application Code** : It is the name that defines an application. An app can be renamed at the List of Application.
  - **Description** : This field contains a brief description of the application objectives.
  - **Documents Path** : The absolute path to store uploaded documents in the application.
  - **Image Directory** : The filesystem directory to store the application images.
  - **Application images** : Import images into the application to allows using them in the application.
  - **Language** : Set the default language of the application. Display all the application hints and messages in the selected language.
  - **Share Location Variable** : Define if the app shares the regional settings with other applications through a session variable.
  - **Charset** : Define a specific charset to use in the application.
  - **Share Theme Variable** : Define if the app shares the Theme settings with other applications through a session variable.
  - **Folder** : Define the project folder that contains the app.
  - **Edit by Project** : Define if other project developers can edit the application.
  - **Timeout** : Set the session runtime timeout in seconds. If the value is Zero, it assumes the default timeout of the PHP.
  - **HelpCase Link** : It allows to associate a HelpCase file with the application.

- **Maximum file size** : Set the max size of uploaded files in the application.
- **Message title** : Message box title.

Notification settings

Error Settings ### {#id-2}

It groups the notification options of the app.

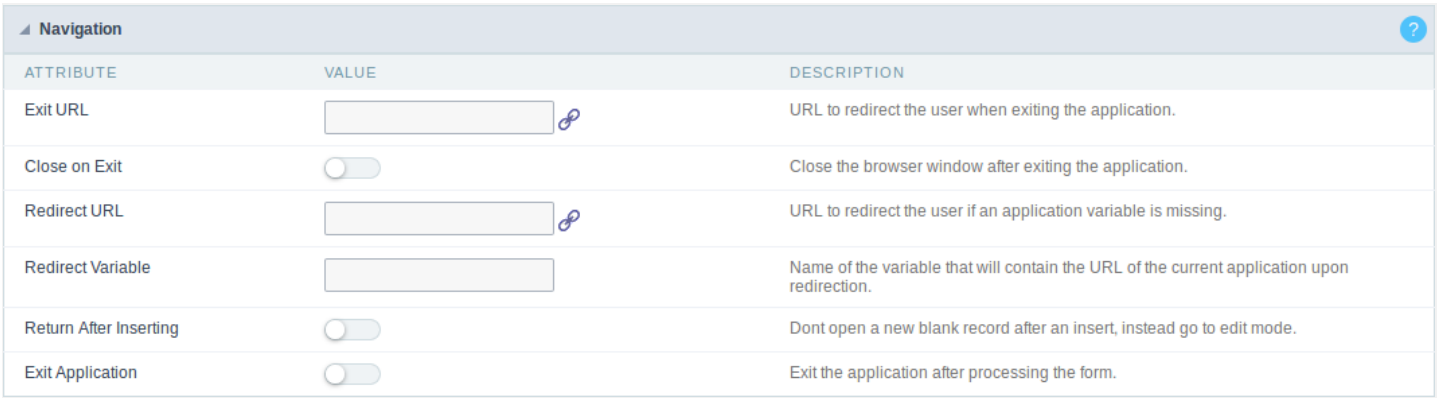


Error Settings Interface.

- **Attributes**
  - **Use SweetAlert**: Use the SweetAlert to display messages from the application. When this option is active, it will replace the browser's "confirm" and "alert".
  - **SweetAlert position using Toast** : The position to display error messages on the application.
  - **Error Position on the field** : The position to display error messages when criticizing the field.
  - **Show the Error Title in the Application** : Define to display the title line of the error message or not.
  - **Show the Error Title in the Field** : Define to display the title line of the error message in the field or not.
  - **Script Error** : Allows displaying the line code where there is an error..
  - **SQL Error** : Allows displaying the SQL statement if it got an error.
  - **Debug Mode** : Runs the application in Debug mode, showing all SQL statements the application is executing.
  - **Ajax Error Output** : Enables the Ajax alert for debugging errors.

Navigation

This interface allows defining the navigating behavior of the application



Navigation Interface.

Exit URL

URL to where the user goes when he clicks on the “exit” button.

Close on Exit

Close the browser window when the user clicks on the “exit” button.

Redirect URL

Redirect to another URL in case there aren’t any global variables available.

Redirect Variable

Creates a variable with the application name and sends it to the redirected application.

Return After Inserting

It opens the inserted record in edit mode instead of to open a new blank form.

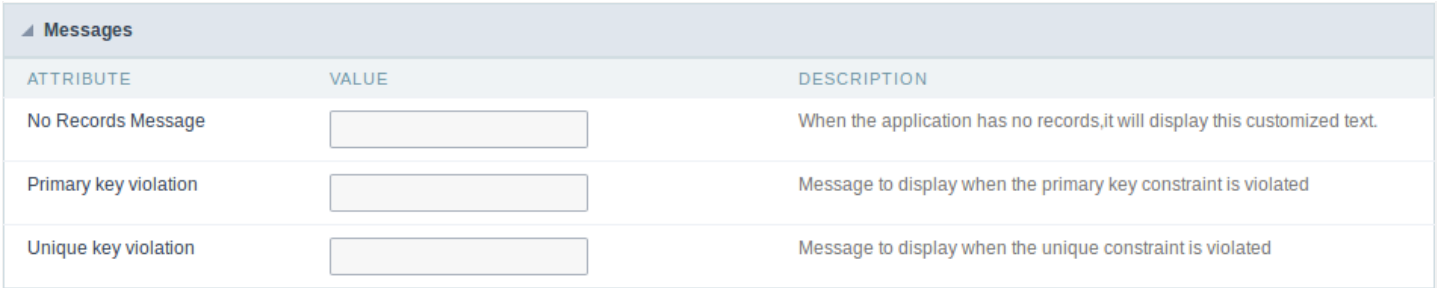
Exit Application

It redirects to another application after inserting, updating, or deleting a record.

Messages

On this screen, you can redefine the default application messages to the end-user by customizing the validation messages for Insert, Update, and Delete. You can also define messages for SQL errors and the confirmation outputs.

Messages



No Records Message

The message displayed when the application has no records.

Primary key violation

The message displayed when there is a violation in the database’s primary key constraint.

Unique key violation

The message displayed when there is a violation in the database’s unique constraint.

Insertion Messages

Insertion messages

ATTRIBUTE	VALUE	DESCRIPTION
Message After Insert		Message to display after inserting a record.
Message to Confirm Insert		Message to display to confirm inserting of a record.

Message After Insert

The message displayed when inserting a new record.

Message to Confirm Insert

The message displayed if the end-user wants to confirm the inserting of a new record.

Update Messages

Update messages

ATTRIBUTE	VALUE	DESCRIPTION
Message After Update		Message to display after updating a record.
Message to Confirm Update		Message to display on the update confirmation

Message After Update

The message displayed when updating a record.

Message to Confirm Update

The message displayed if the end-user wants to confirm the changes of a record.

Messages of delete

Messages of Delete.

ATTRIBUTE	VALUE	DESCRIPTION
Message After Delete		Message to display after delete a record.
Message to confirm Delete		Message to display on the delete confirmation

Message After Delete

The message displayed when deleting a record.

Message to confirm Delete

Displays a customized message asking to confirm the record deletion.

If you make no changes, the application will use the default values for the messages. Those values can be configured in Locales -> Application Language.

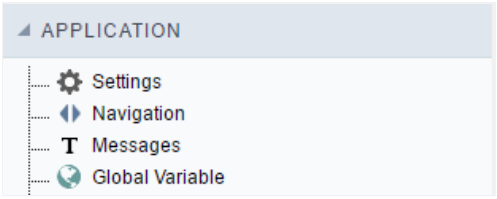
Global Variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**"**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target="_blank"} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



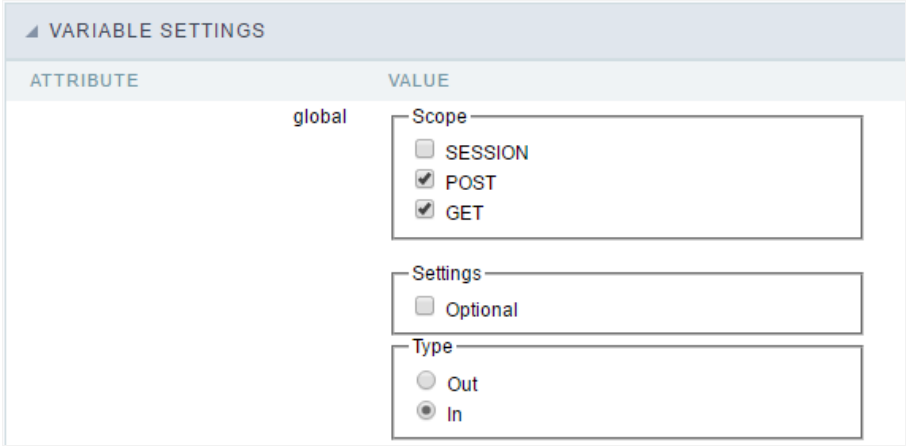
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

Indicates where the application is using the variable.

Synchronize table

This process will performs a comparison between the definitions of application fields and the connected database fields. If there is any change within the connected table, such as adding a new field, exclusion, or a change in the data type, the changes will appear visible like in the image below:

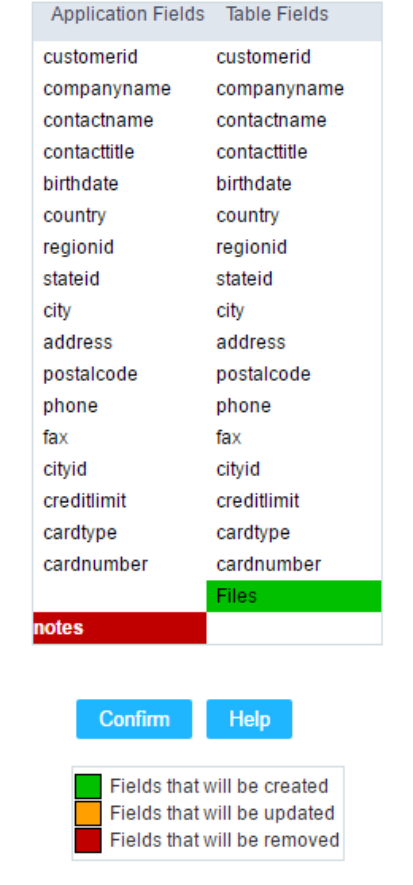


Table synchronization interface.

After accessing the function from the link "Synchronize table" you will see a comparison table between the application fields (left table) and the database table fields (right table), like the image above, the fields highlighted in "red" will be deleted of the application, the highlighted fields in "Green" will be inserted in the form, and the highlighted fields in "orange" will be updated, so data type will be updated.

To rename a field in the database table, the table synchronization effect, by comparison, will be the same as deleting a field that existed in the application and the inclusion of a new field.

Related Links 🔗

Related Videos ▷

Calendar Links Overview

This feature allows the developer to create links between applications of the same project, expanding the integration of applications. All link options are grouped under the Application Links menu.

Application Links

In the first access to the menu, we can see the list of existing connections in the application.

> If the application does not have a configured link, the application list screen will be displayed with the message: **This application does not have any link. Click here to create one now.**

ID

Link identification ID.

Type

The type of link created, some links such as Edit Link allow only one link per application. In this case, the developer will be able to check the connection types that already exist in the application

Target Application

This info show the target application name.

Actions

This column has edit options for the links.

Properties

It allows accessing the binding properties where it is possible to configure the binding behavior.

Link

Displays the links screen, where it is possible to configure the link that was made with the application informed in the target application column. In this option it is possible to change the parameter passed in the connection as well as the target application.

Delete

Permanently deletes the connection in the applications.

> Through the option Restore Applications, it is possible to get a previous version of the application, making it possible to recover the link deleted.

Links Type

The Calendar application has the following links options

- Field Link: Allows you to create a link through the application fields to any application in the project.
- Capture Link: It allows the creation of links from the filter fields of the Grid application, in order to enable the recovery of the value to fill in the field, with another Grid application of the project.
- Button Link: It allows creating a link through the buttons created by the developer in the query application, to any application in the project.

Calendar Applications Link Types x Target Application available to link

Check the links types of the Calendar application and the target applications available for each link types.

Only **Capture link** have some restrictions in the target application use, allowing only to use of Grid application.

	Field Link	Capture Link	Button Link
Grid			
Chart			
Form			
Control			
Search			
Menu			
Tree Menu			
Tabs			
PDF Report			
Dashboard			
Blank			
Calendar			

Related Links

- Google Calendar integration
- Contact Form template
- Calendar per User
- Responsive calendar
- Calendar overview
- Google calendar integration
- Form overview

Related Videos

Calendar Field Link Settings

Creating a field link

Allows to create a link, represented through a link, from a grid’s column to any project’s existing application. All the links are displayed in a dropdown if there are more than one link to the same field.

In the types of link option, we will choose the **Field link**.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project’s applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

![Same window][mesma_janela]

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open the application in a tab inside the menu:

Choosing the option **Yes**, the application will be open in a **menu tab**, when executed from a menu application, instead of opened in another browser window.

Open in an iframe

When we use this option the target application will be displayed in the same window of our application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Iframe properties

In this settings, we can define some iframe details that will display the target application.

Iframe position in relation to the main application:

In this option, we will define in which position in relation to the main application the iframe will be displayed, there are four options:

Below: The iframe will be displayed below the main application.
Above: The iframe will be displayed above the main application.
Right: The iframe will be displayed on the right of the main application.
Left: The iframe will be displayed on the left the main application.

Iframe height:

Allows to set the iframe height.

Iframe width:

Allows to set the iframe width.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Grid’s Properties

In those settings we can define how our target grid will be displayed. We inicially have six options, that are:

![Form proprieties][propriedades_consulta]

Initial module:

We can define how our application will be executed in those two options:

Grid: The application will be executed as a grid.
Search: The application will be executed as a search.

Quantity of rows:

In this option we can define the quantity of rows that will be displayed in the target application.

Quantity of Columns:

In this option we can define the quantity of columns that will be displayed in the target application.

Pagination:

In this option we can define if the target application’s pagination will be total or partial, in case partial is the selected option the quantity of rows will be the informed previously.

Enable header:

In this option we can define if the target application’s header will be displayed.

Enable Navigation buttons:

In this option we can define if the buttons **first, previous, next and last** will be available in the target application.

Calendar Capture link Settings

Creating a Capture Link

The capture link is used to return a value from a **Grid** to a **Form** field.

In the types of links options, we will choose the **Capture Link**. After selection this option, we should also choose which field we want to return the value of.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

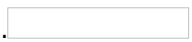
We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side. 

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Application with no parameters

However, when the target application does not have any defined parameters, the following screen is displayed:

Clicking in the button, you will be taken to the target application to create a parameter, so you can use the update button in the **parameters definitions** to refresh them.

Link properties

In this screen we will set the application display mode that will be called in the link.

In this type of link there is only one display option:

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Modal:

In this option we are going to define if the target application will be opened in a Modal.

Yes: This option will make the target application be opened in a modal. **No**: This option will make the target application be opened in a new window.

If **Yes** is selected in the previous option, the Modal **Height** and **Width** will be available.

![Modal with yes][modal2]

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

If **No** is selected in the previous option, only those options will be available.

Allows to modify manually in the update:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Allows to modify manually in the insert:

Indicates which shortcut key to the button add new register.

Allows to modify automatically in the update:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Allows to modify automatically in the insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Related Links

- Google Calendar integration
- n template
- User
- Responsive calendar
- Calendar overview

Related Videos

- Google calendar integration
- Form overview

Calendar Edit Link Configuration

In the Link Folder of the Application Menu (Image Below) are displayed the links existing in the application and also the item New Link. When clicking on the existing link it is displayed the screen below that allows to manage the links.

Editing Links.

Actions

Properties	Change the link’s behavior, position, and how the link opens.
Link	Change the application that’s being called in the link their parameters.
Delete	Remove the existing link.

Related Links

- Google Calendar integration
- n template
- User
- Responsive calendar
- Calendar overview
- Google calendar integration
- Form overview

Related Videos

Calendar Button link Settings

Creating a Button Link

Allows the developer to create a link where the call to the other application will be done through a button.

In the type of links options, we will choose the **Button Link**. Choosing this option it will be possible to create a link to any other application.

List of applications

After selecting this option, The list of applications to what you want to create a link will be displayed.

This screen can be viewed in the following ways:

All:

In which is possible to see all the project’s applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Link properties

In this screen we will set the application display mode that will be called in the link.

There are five display options, they are:

Open in the same window:

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

Open in another window:

When we use this option the target application will be displayed in other browser window, and the target application will have a exit button so we can close this window.

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Open in the same window

When we use this option the target application will be displayed in the same window of the application, and the target application will have a back button so we can return to the previous application.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open in another window

When we use this option the target application will be displayed in another browser window.

When we select this option, we can set the following options:

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Open the application in a tab inside the menu:

Choosing the option **Yes**, the application will be open in a **menu tab**, when executed from a menu application, instead of opened in another browser window.

Choosing the option **Yes**, there are also some other settings to be displayed:

Title to the tab:

This option allows the developer to set a title to the tab that will be opened when used in a Menu application.

Hint to the tab:

This option allows the developer to set a message to be displayed when the mouse cursor is over the tab Menu.

Active tab icon:

This option allows the developer to set an icon to be displayed in the tab when used in a menu application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Height:

Allows to set the Modal's height.

Width:

Allows to set the Modal's width.

Exit URL for the target application:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Close the form after an update:

Choosing the option **Yes**, the window will be closed after doing the update of a register.

Close the form after an insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Hint for the link:

In this option we will inform a message that will be displayed when the mouse cursor is over the button with the link.

Related Links

- Google Calendar integration
- n template
- User
- Responsive calendar
- Calendar overview
- Google calendar integration
- Form overview

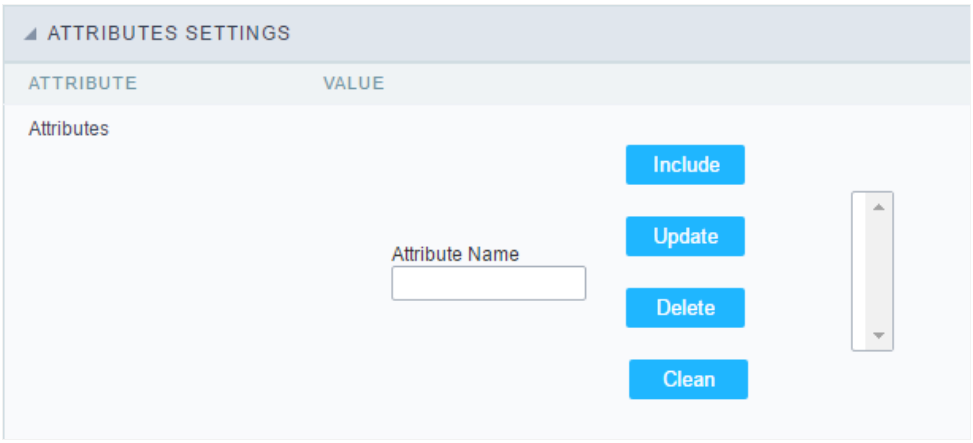
Related Videos

Calendar Programming

The concept of programming is incorporated in this ScriptCase version. With the use of resources of attributes, methods and libraries. In the previous version was already possible create business rules in the applications, using this concept. The biggest difference is that now this can be realized in a more organized way, making it easier to be developed and comprehended by another developer.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal Libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

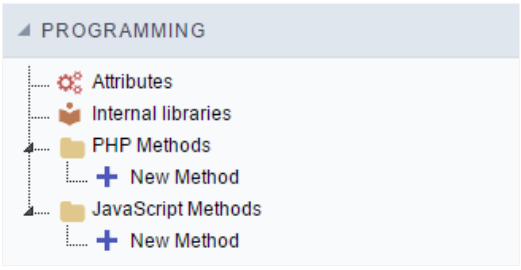
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

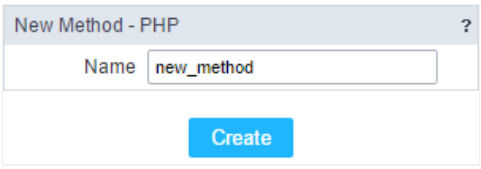
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

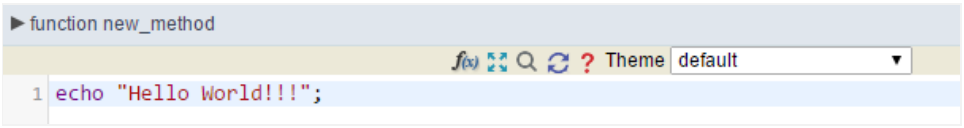


- Give the method a name and click Create. As in the image below.

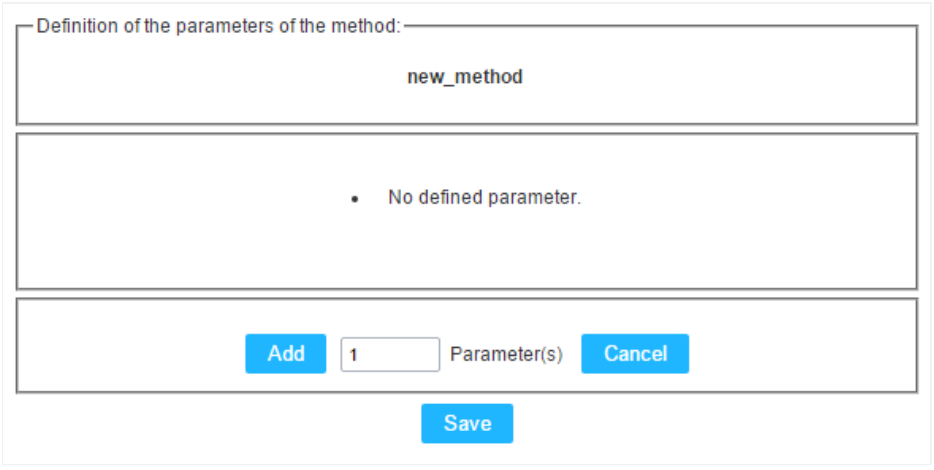
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



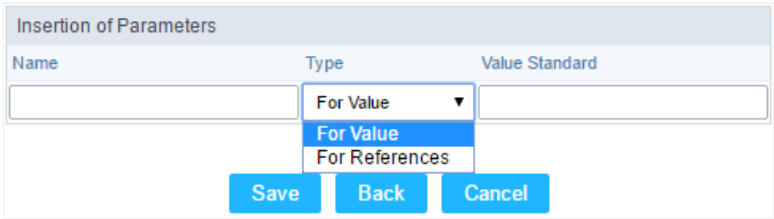
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

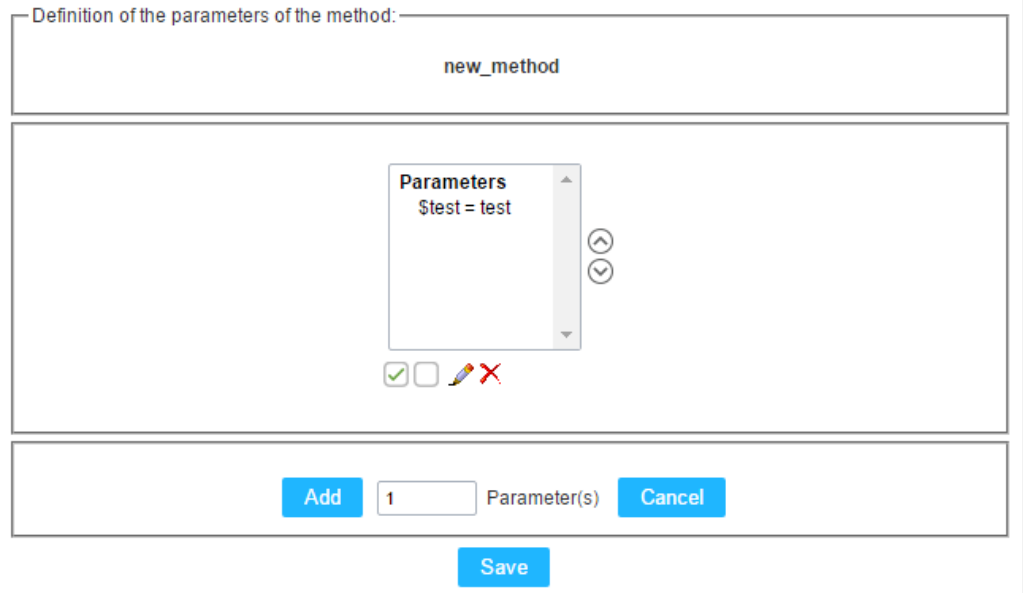


- **Name** : Type in the variable's name.

- **Type** : Selecting the type of variables: For Value or For Reference.
- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:

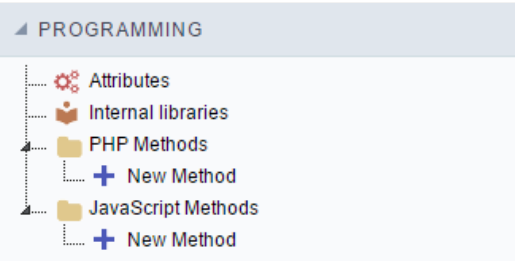


- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

JavaScript Methods

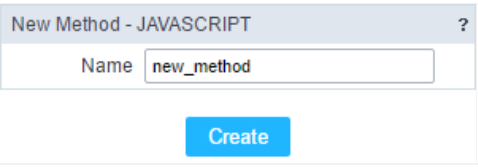
Methods are function or procedures declared by the developer, that helps when appling the routines. Using methods in the applications allows to reuse of the your code throughout the application, optimizing the development experience.

Creating a new method



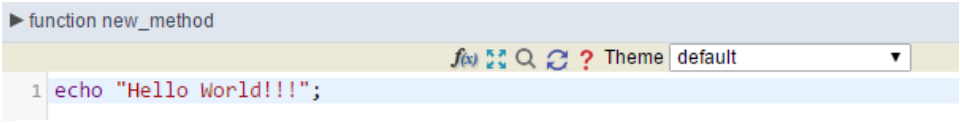
JavaScript method creation Interface

- Define a name for the method and click on Create. Like the image below.

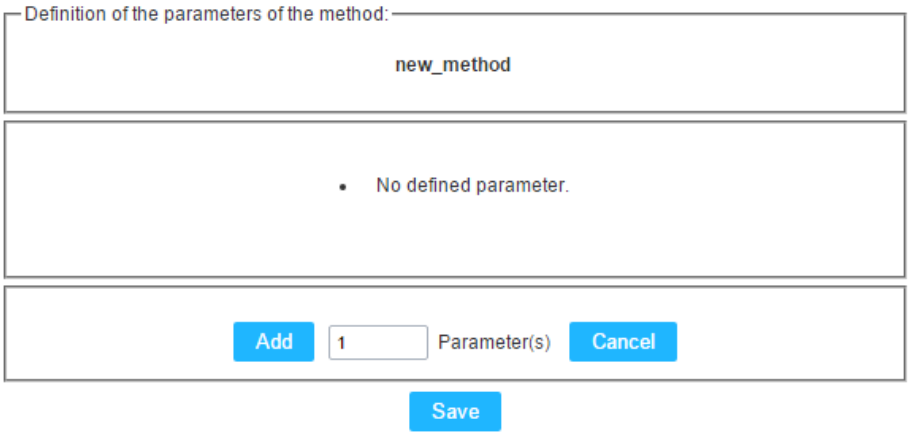


Include Method.

- Methods can receive parameters.

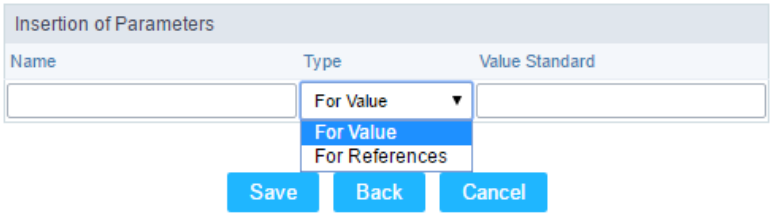


  - Add the amount of variables:



JavaScript method parameter definition

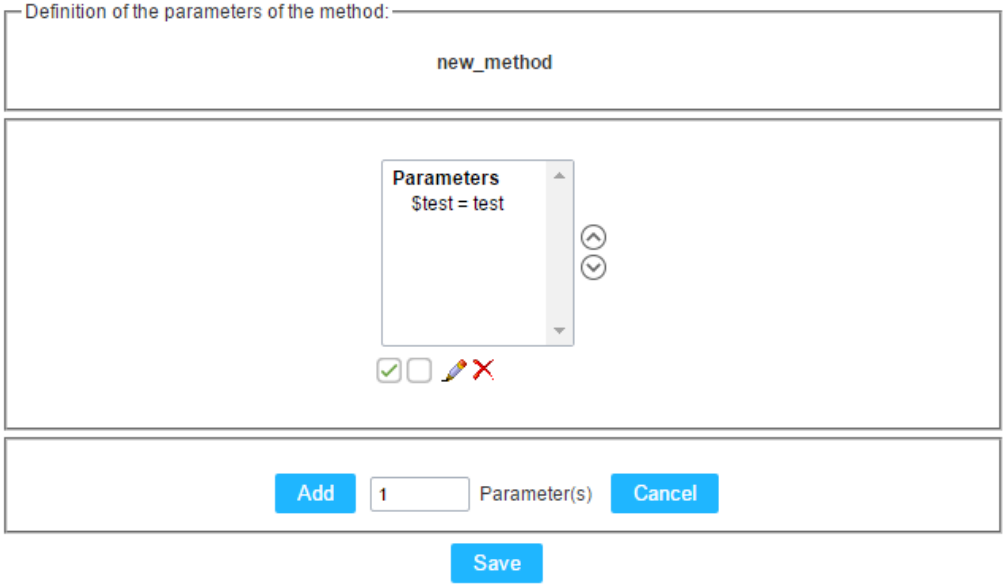
  - Defining the variables:



Defining the variables

    - **Name** : Type in the variable's name.
    - **Type** : Selecting the type of variables: For Value or For Reference.
    - **Value Standard** : The parameter's value used to initialize when calling the method.
      References allows to create a second name for a variable that you can use to read and modify the original variable's information.

  - Editing a parameter:



    - : Checking all the parameters in the parameter list.
    - : Uncheck all the parameters in the parameter list.
    - : Edit the selected parameter of the list.
    - : Deletes the selected variable of the list.

Related Links

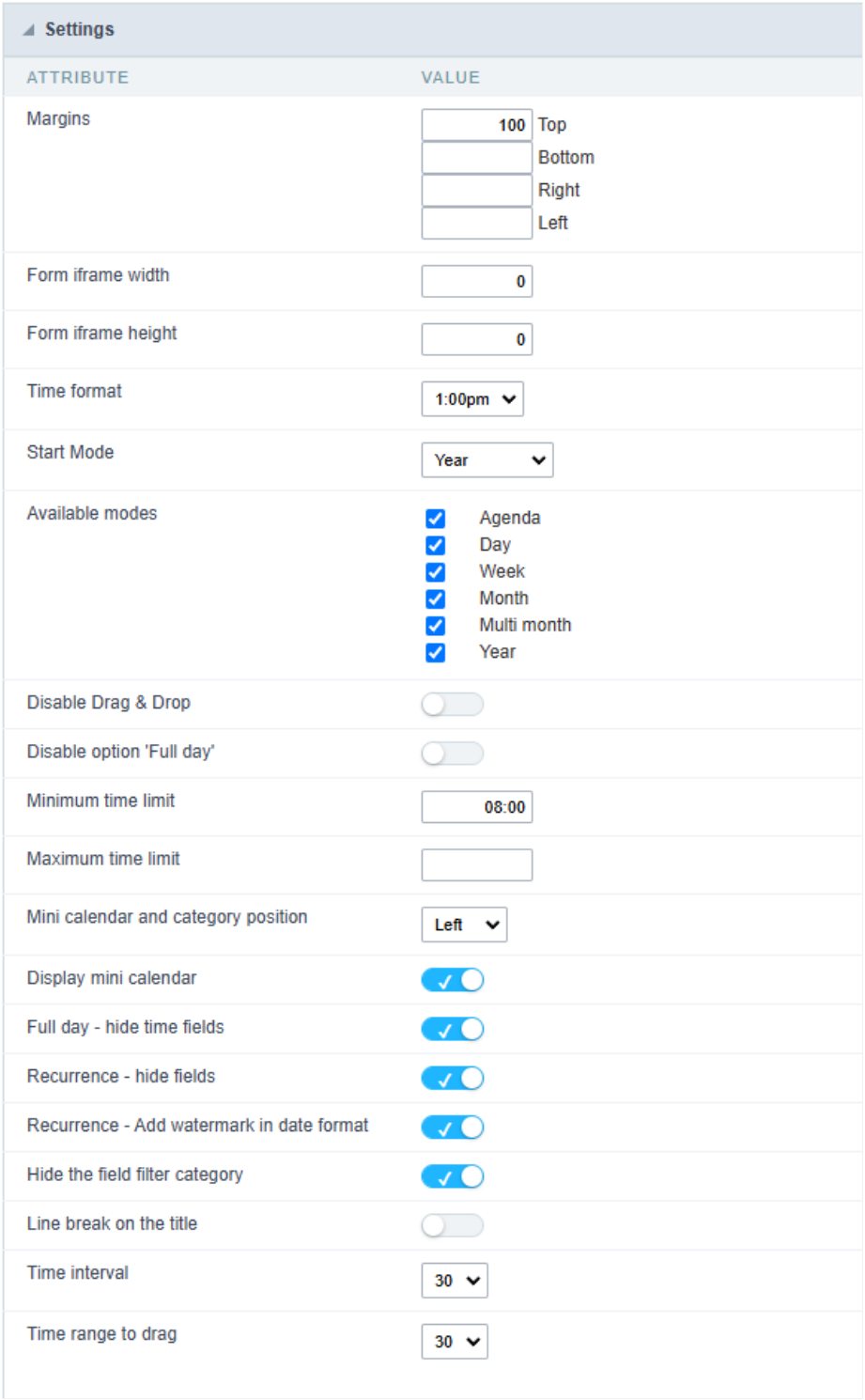
Related Videos

Calendar Settings

The **Calendar Settings** screen allows you to customize various aspects of the application, ensuring better adaptation to project needs.

Through these settings, the developer can adjust the calendar's behavior, choose which fields will be displayed, define rules for event creation and editing, and configure colors and styles for better visualization.

Settings



Calendar Settings Screen

Margins

Allows the developer to set the application's margin values in pixels. If no value is specified, the application will use the default values from the configured theme.

Attention:
When configuring the **Vertical Alignment** attribute, the defined margin values will be ignored.

Example of how to use the attribute

Form Iframe Width

Sets the width, in pixels, of the form used to insert events in the **Calendar**.

Form Iframe Height

Sets the height, in pixels, of the form used to insert events in the **Calendar**.

Time Format

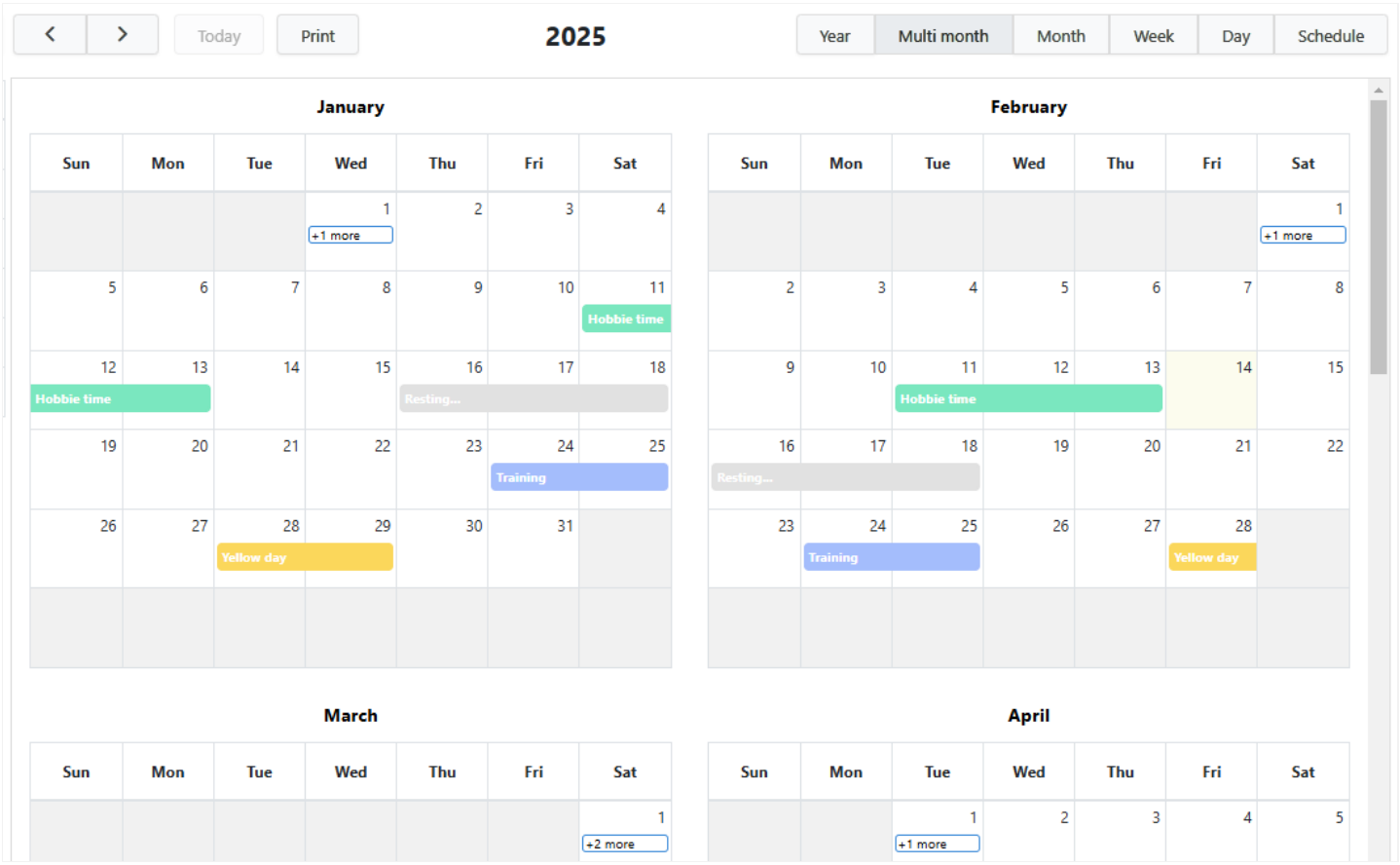
Defines whether the time display will follow the **12-hour** (AM/PM) or **24-hour** format, affecting how events are viewed and entered.

- **13:00** - Uses the **24-hour** format.
- **1:00 PM** - Uses the **12-hour** format (AM/PM).

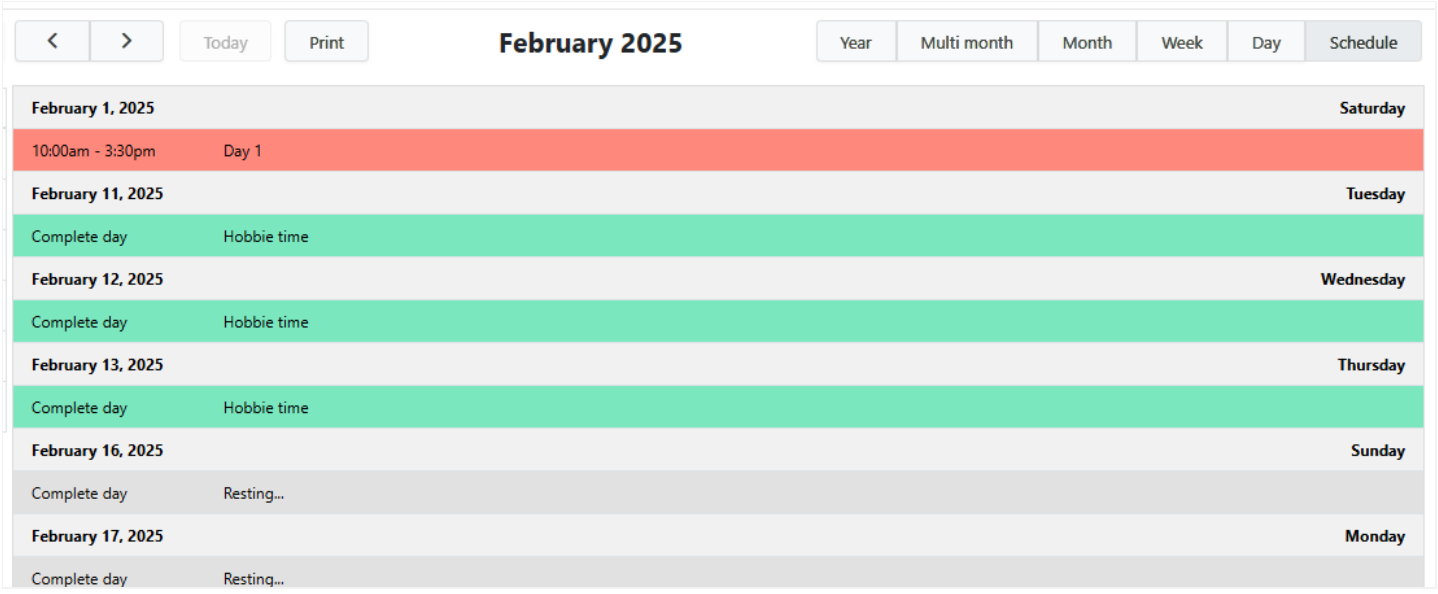
Initial Mode

Defines the default display mode of the calendar, which can be set to: **Agenda**, **Day**, **Week**, **Month**, **Multi-Month**, or **Year**.

Example of Multi-Month View



Example of Agenda View



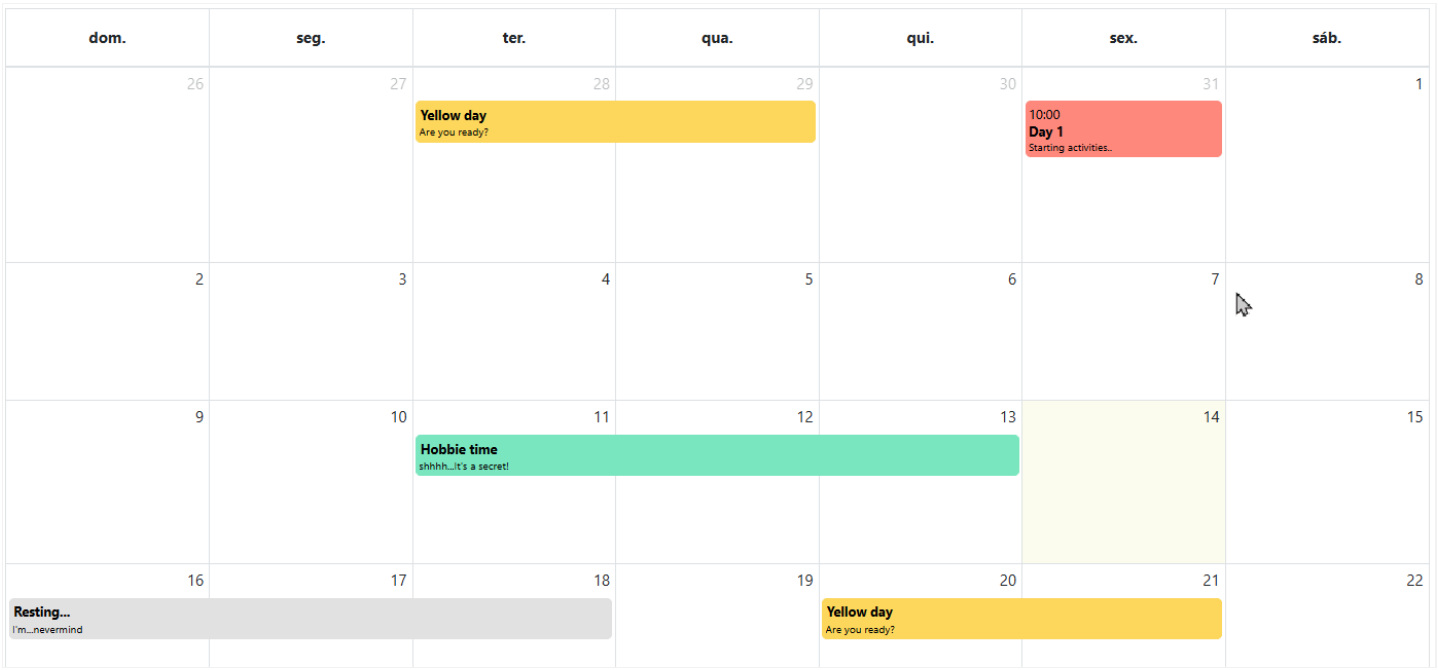
Available Modes

The developer must specify which viewing modes will be available to the end user by deselecting those they do not wish to offer.

Disable Drag & Drop

Disables the Drag & Drop feature for calendar events. Drag & Drop allows updating event dates and times without accessing the edit form.

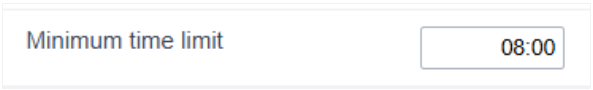
Example of Drag & Drop Usage



Disable ‘All Day’ Option

Disables the option to set an event as “All Day,” removing the field from the event creation form.

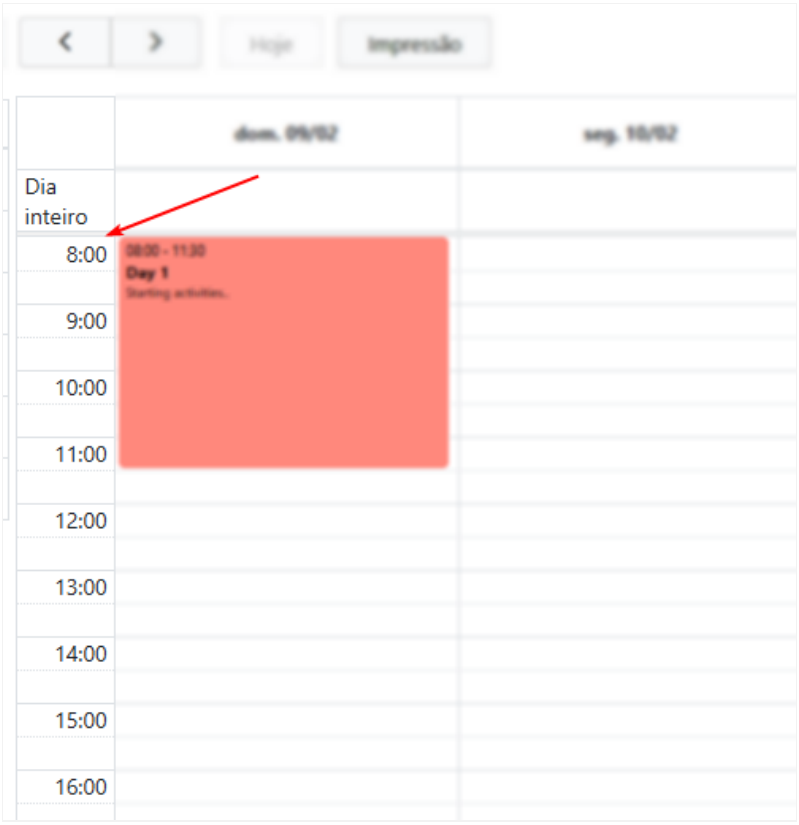
Minimum Time Limit



Sets the earliest time displayed in the **Week** and **Day** views of the calendar.

Important:
The specified time must follow the **hh:mm** format, as shown in the image; otherwise, it will not be considered.

Example Using Initial Time Limit



Maximum Time Limit

Sets the latest time displayed in the **Week** and **Day** views, limiting the time available for event creation.

Important:
The specified time must follow the **hh:mm** format, as shown in the image; otherwise, it will not be considered.

Mini Calendar and Category Position

Determines the position of the **Mini Calendar** and **Categories** display column, which can be placed on the left or right side of the main calendar.

The column is displayed when the mini calendar or event categories are in use.

Display Mini Calendar

Enables the Mini Calendar, which allows independent navigation from the main calendar.

All Day - Hide Time Fields

Hides the start and end time fields when marking an event as “All Day” in the event creation form.

If this option is disabled, any values entered in the time fields will be ignored when marking an event as “All Day.”

Recurrence - Hide Fields

Hides the recurrence period and information fields in the event creation form of the **Calendar**.

Recurrence - Add Date Format Watermark

Displays a **watermark** in the recurrence date field, helping users identify the expected date format.

Hide Category Field Filter

Hides the **Category** field filter, preventing users from selecting specific categories for display in the **Calendar**.

Line Break in Title

Enables automatic line breaks in event titles when the text exceeds the available space.

Time Interval

Defines the time interval, in minutes, used to structure events in the **Week** and **Day** views.

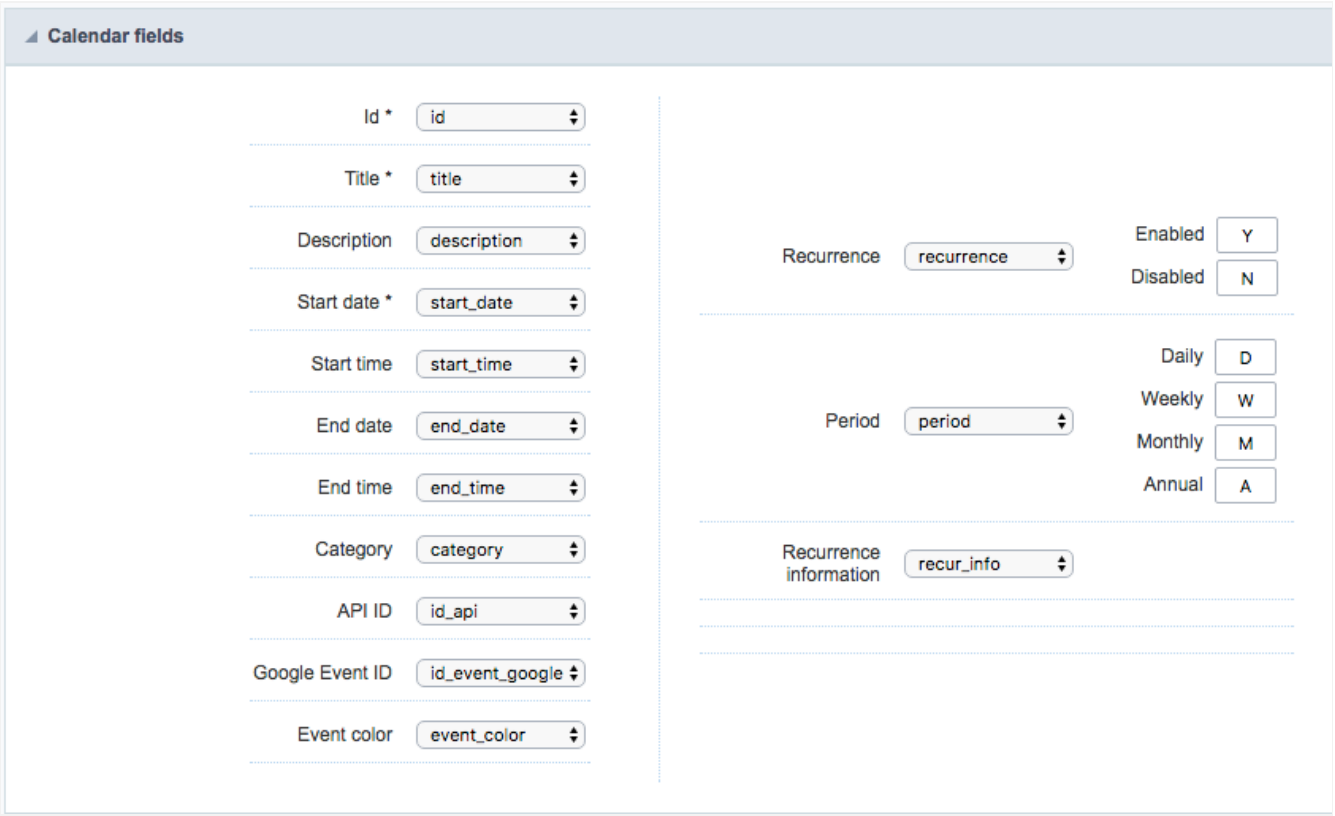
Time Interval for Dragging

Specifies the time interval applied when creating or moving events in the **Day** and **Week** views. If this value is greater than the one set in the **Time Interval** attribute, it will automatically adjust to the lower value.

Calendar Fields

This interface displays the fields relationship that will be used in the application.

You need to relate the table fields to the form application fields.



Calendar Fields Settings Screen

Google API

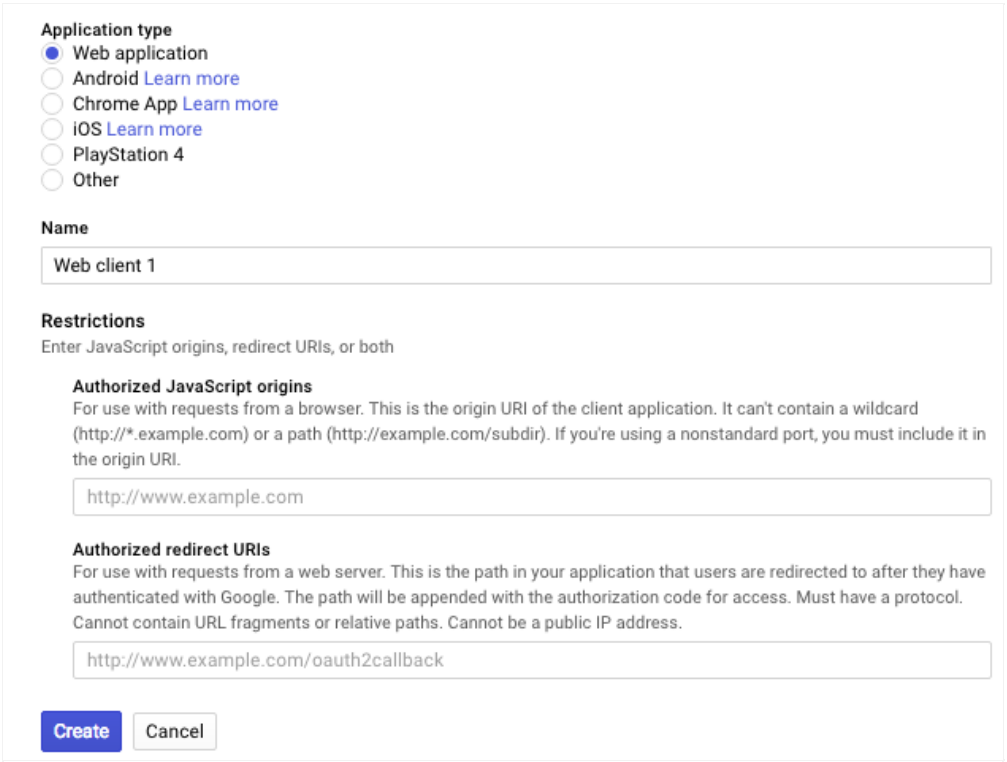
The Google Calendar API will allow you to synchronize your Google calendar with the application developed in Scriptcase.

First you must activate the Google Calendar API by clicking on https://console.developers.google.com/apis/api/calendar-json.googleapis.com/overview.

After activation you need to configure the “OAuth consent screen” by clicking on

https://console.developers.google.com/apis/credentials/consent

.



OAuth consent screen

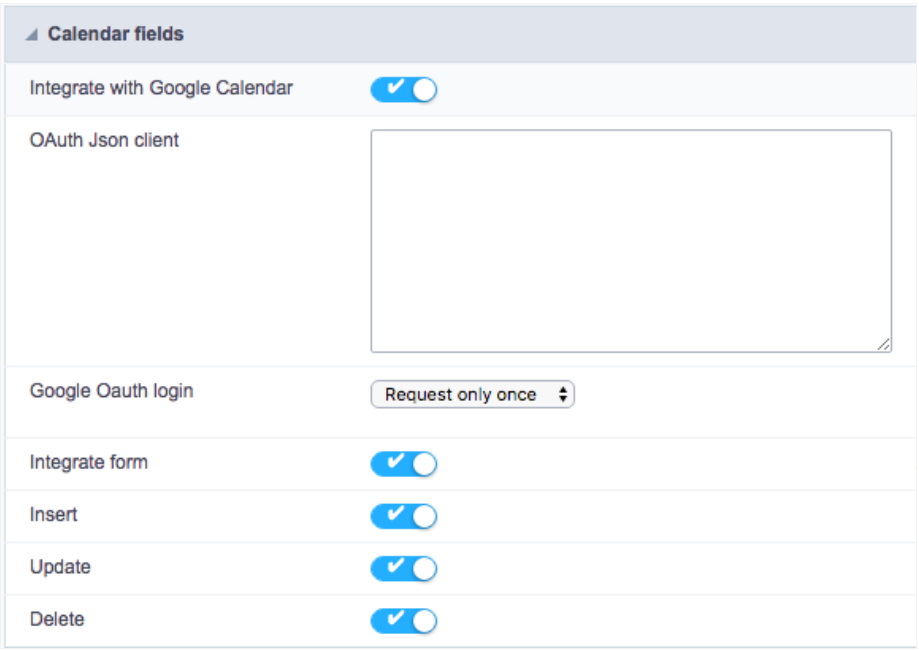
- **E-mail Address** : Select the email that will be used as the source for the google calendar. The application will use this data for synchronization.
- **Product name shown to users** : Allows you to enter the consent name that will be displayed to the clients. Make sure the name accurately describes your product and be careful not to use names that suggest that the product is from Google or another company.
- **Home URL** : Allows you to enter a URL to homepage. (optional)
- **Product Logo URL** : Lets you enter a URL that contains an image that will serve as the product logo. (optional)
- **Privacy Policy URL** : Lets you enter the URL of the product privacy policy. (optional)
- **Terms of Service URL** : Lets you inform the URL of the terms of service of the product. (optional)

Note: You only need to inform the e-mail address and the product name. All other items are optional.

Now you need to create a

Credential> OAuth client ID

Select the “Web Application” option and follow the setup below.

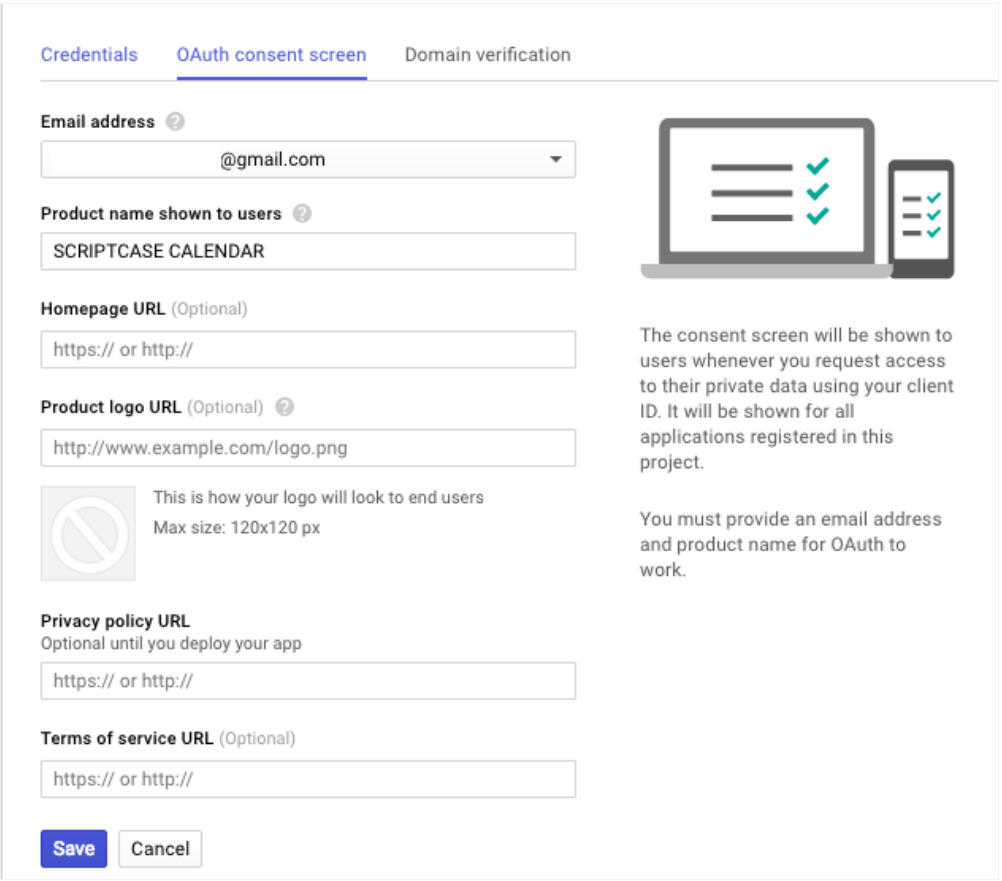


OAuth Client ID Configuration Screen

- **Name**: Enter a name for the OAuth client ID.
- **Authorized JavaScripts Origins** : Insert the root URL of your system. For example: http://yourdomain.com or http://localhost
- **Authorized redirect URLs** : Enter the full URL until your calendar application.

After entering the above data, click “Save.”

Now let’s configure the Scriptcase application.



Google Calendar API Settings Screen

- **Integrate with Google Calendar**: Synchronize with Google calendar events, allowing you to import or export events.
- **OAuth Client Json** : Json from Google OAuth with the credential to access the project. To get this content, open the **OAuth Client ID** of the previously configured OAuth__ client and download the JSON.
- **Login on Google OAuth** : Choose if you want to store the login made in the application to not ask again or if you will request session authentication.
- **Integrate form** : Enables the form to insert, delete or update events in Google Calendar.
- **Insert** : When inserting an event in the form, it also inserts in google calendar.
- **Update**: When updating an event on the form, it also updates in google calendar. * **Delete**: When deleting an event in the form, it also excludes in the google calendar.

Related Links 🔗

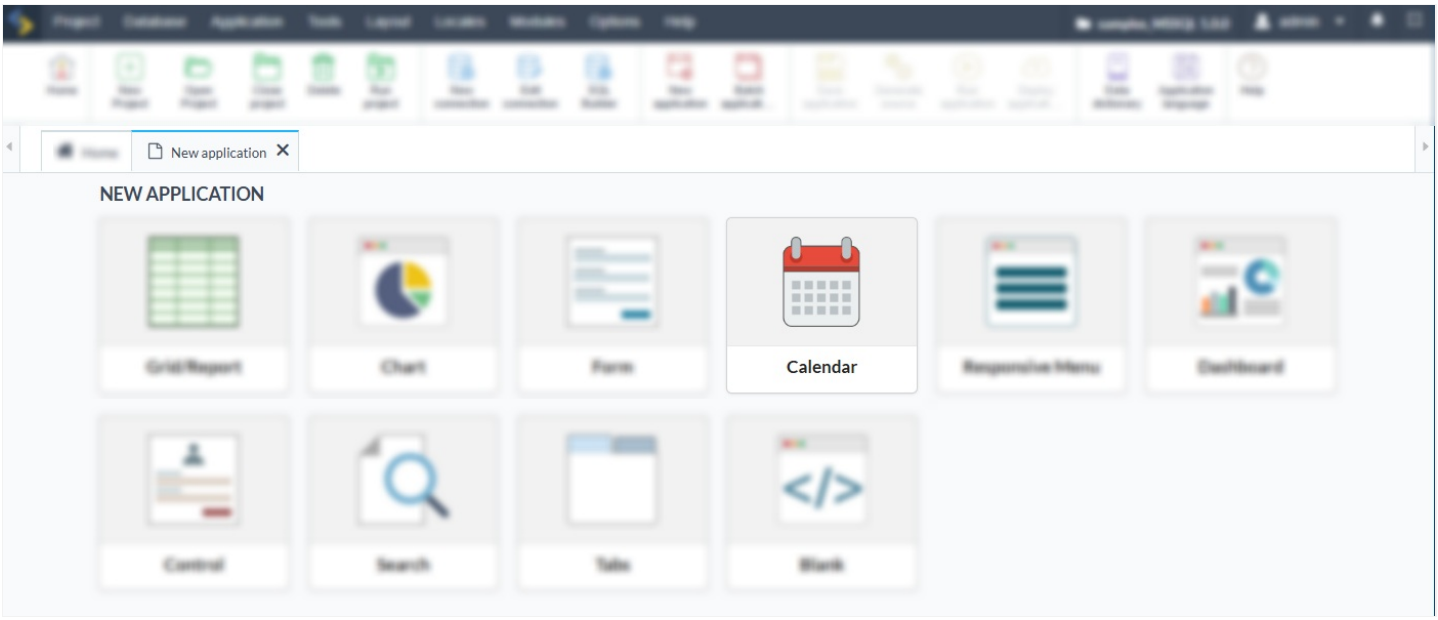
Related Videos ▷

Creating a Calendar Application

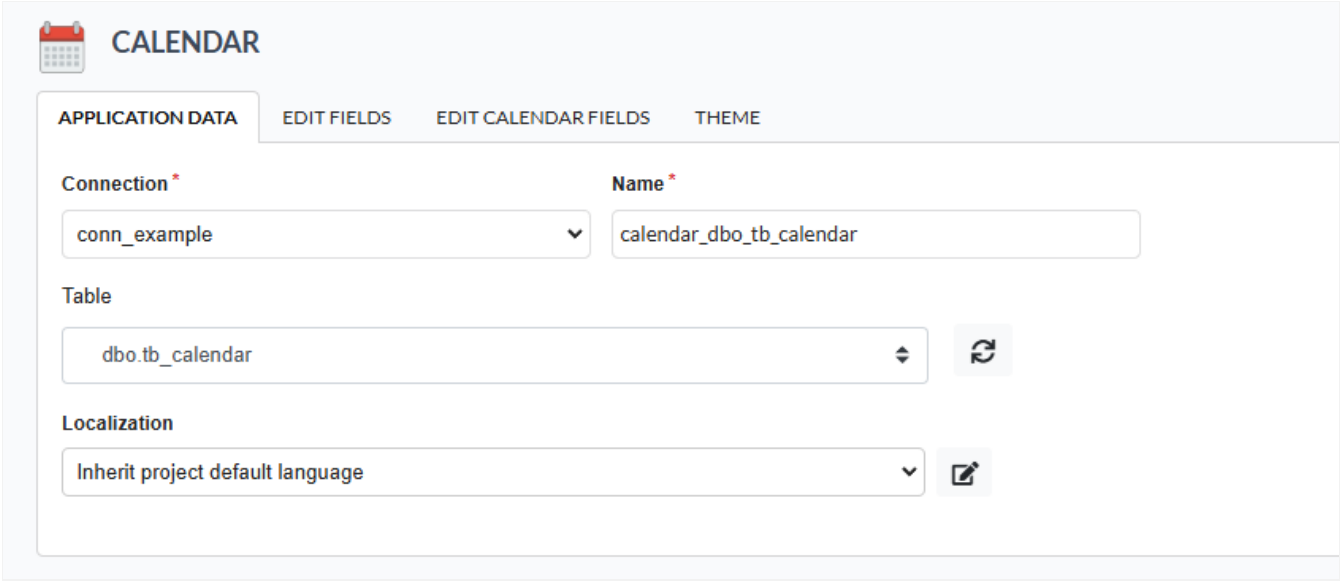
New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.



Application Data



Connection

Defines the connection used for the creation of the application. The existing connections in the project will be listed.

Name

The name of the application being created. It cannot contain special characters.

Table

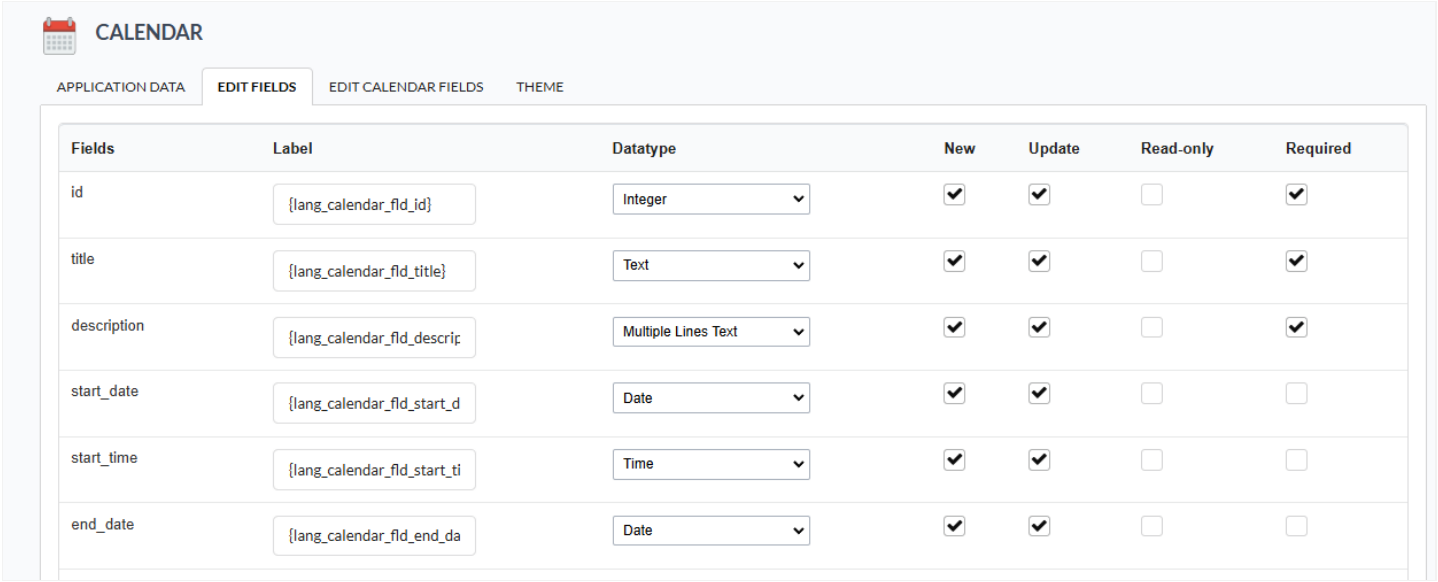
Defines the tables to be used in the application. (Form and Calendar can only use one table).

Localization

The language of the application to be created. The project's default language is automatically selected.

Edit Fields

This screen displays the fields of the selected tables and allows adjustments to be made before creating the application, such as changing the data type, display name, and other configurations.



Fields

Names of the database fields.

Label

Names of the fields in the generated application's interface.

Datatype

Specifies the field's data type.

New

Defines the fields available for insertion.

Update

Defines the fields available for the Update.

Read-Only

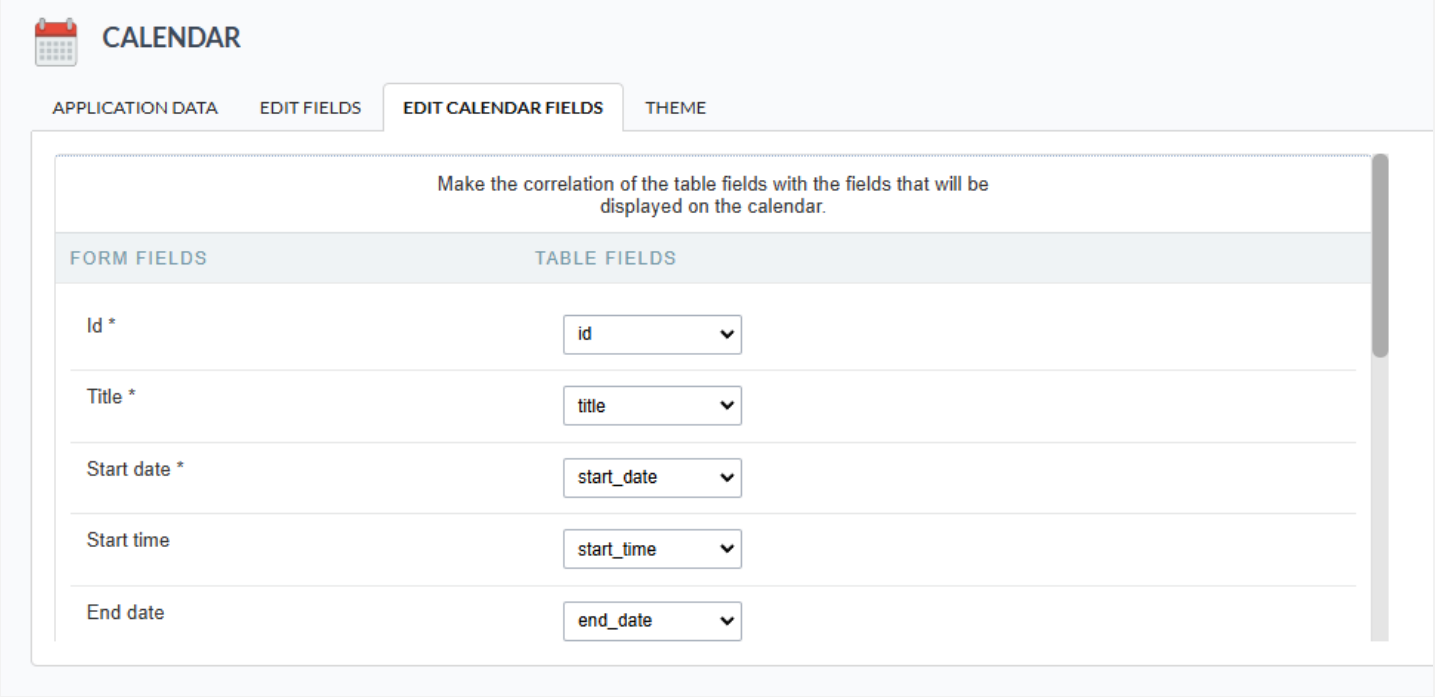
Defines whether the field will be read-only.

Required

Defines whether the field will be mandatory.

Edit Calendar Fields

In this field configuration, we will link the calendar fields to the fields of the table being used in this application.



Form Fields

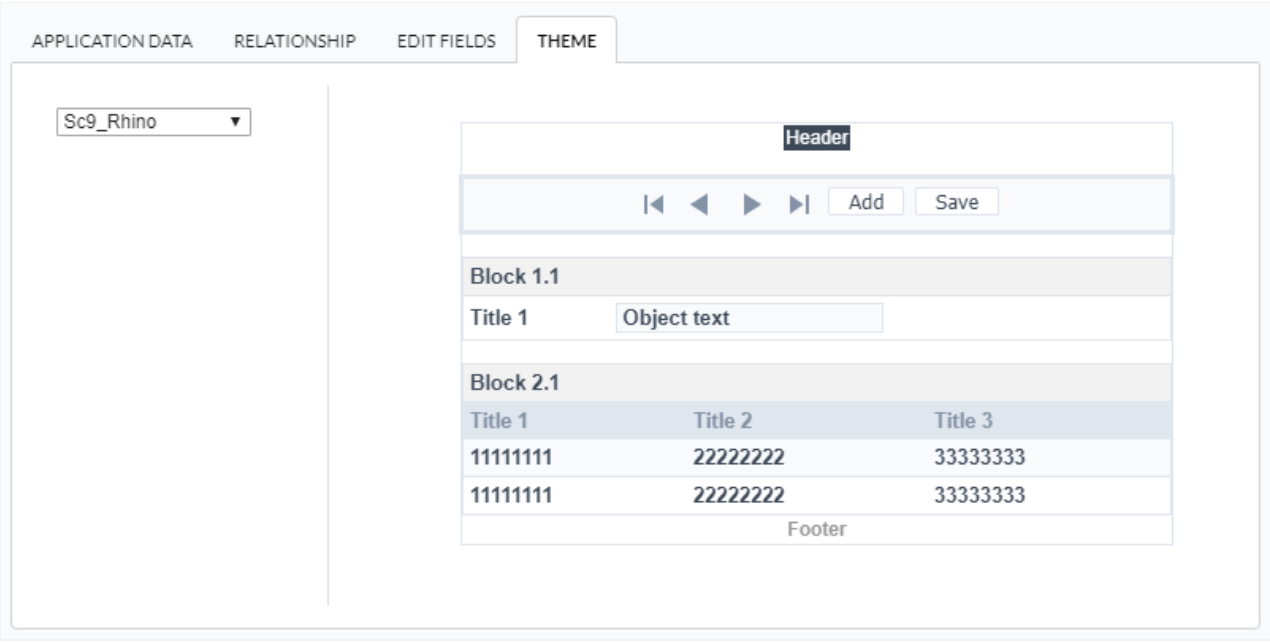
These are the fields returned from the database table.

Table Fields

The calendar fields that will be linked to those in the table.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and modified in **Project** > **Properties**, is automatically selected.



Related Links 🔗

Related Videos ▷

Calendar Mobile Optimization

The Mobile settings allows to define the automatic optimization of generated applications to run on mobile devices.

See below the available settings.

Mobile optimization

Enable mobile optimization

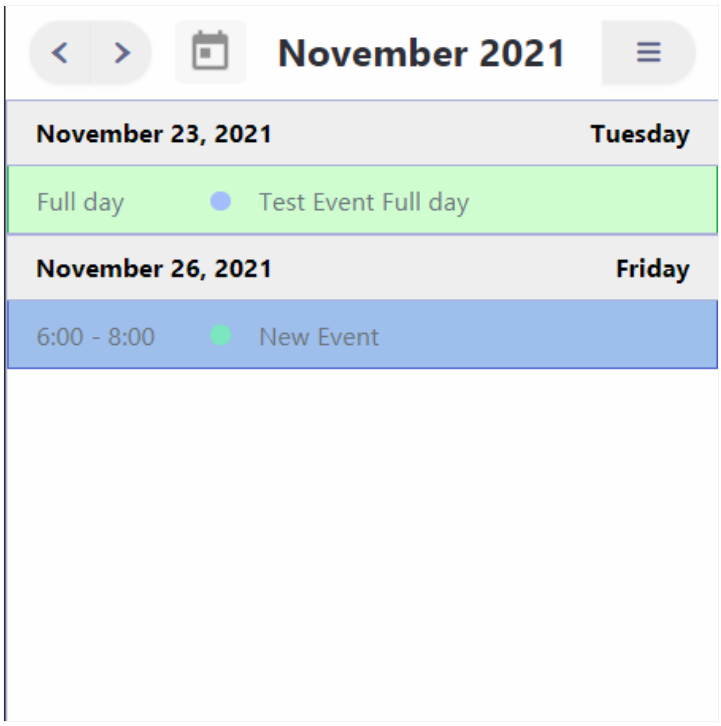
This option changes how the application HTML elements works, adapting them automatically to run on mobile devices.

It enables this option by default when creating a new application.

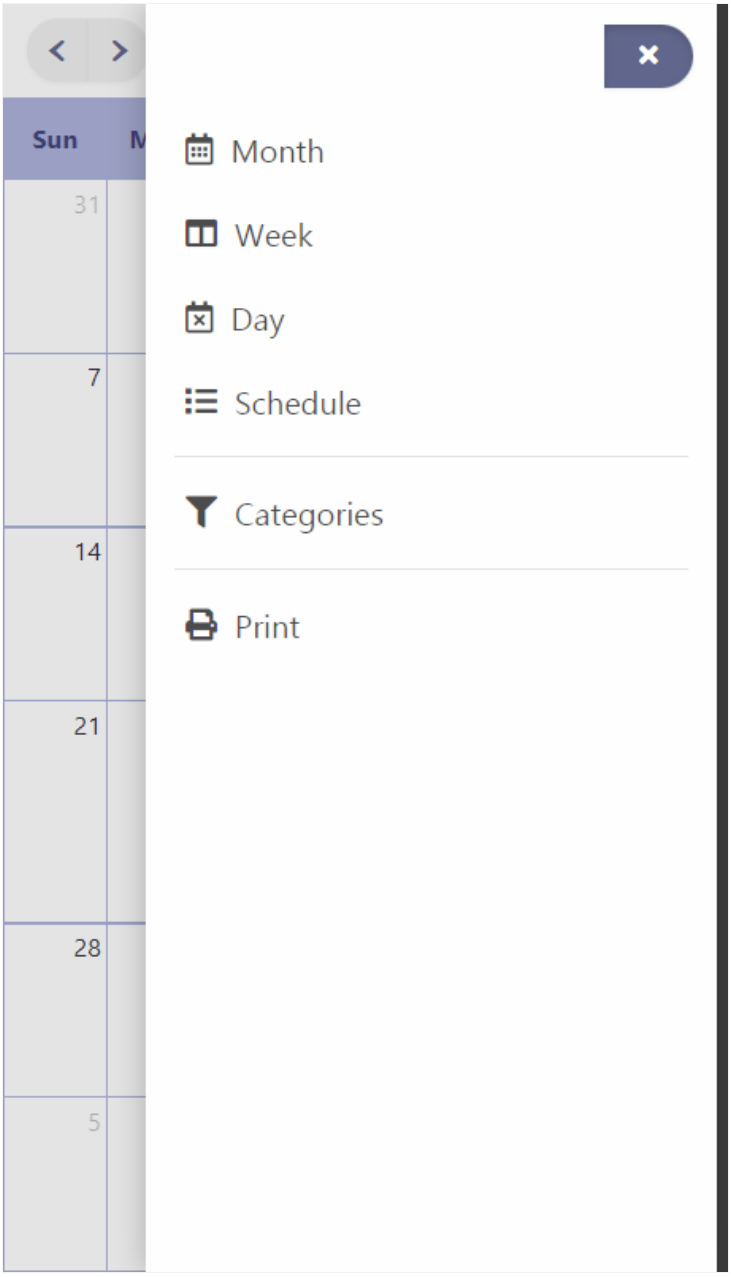
See some examples of adapted screen:

Preview example by month

Sample view in schedule mode

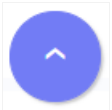


Example of with expanded view menu



Enable back to top button

It enables the displaying of the button **back to top button** in the Grid, Filter, Detail and Summary modules.



It enables this option by default when creating a new application.

Back to top button position

Defines the position of the **back to the top** button:

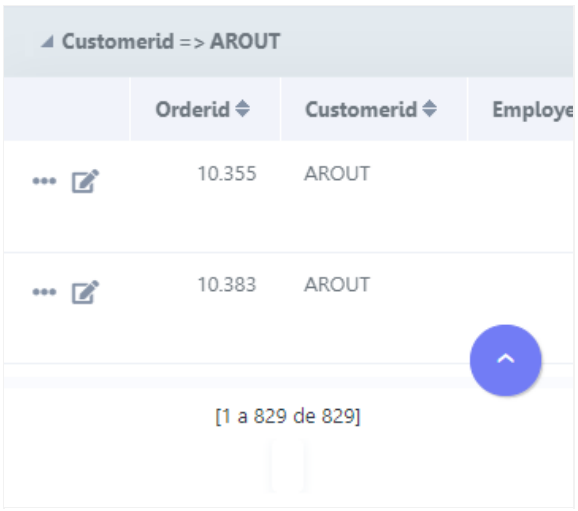
The options are:

- **Right** - Position the button in the bottom right corner;
- **Left** - Position the button in the bottom left corner;

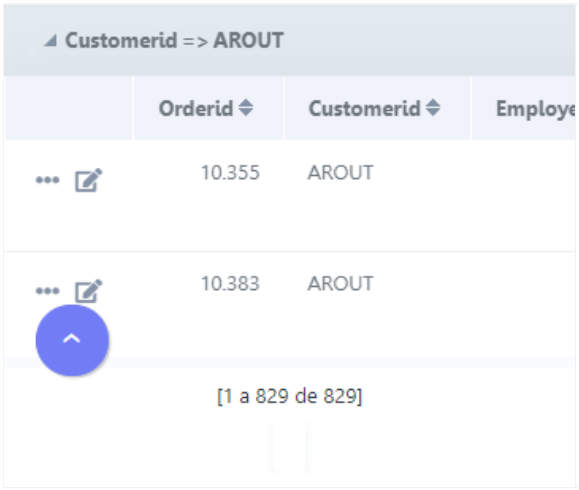
When creating a new application, the button is configured on the right by default.

See below for placement examples.

Button in the bottom right corner



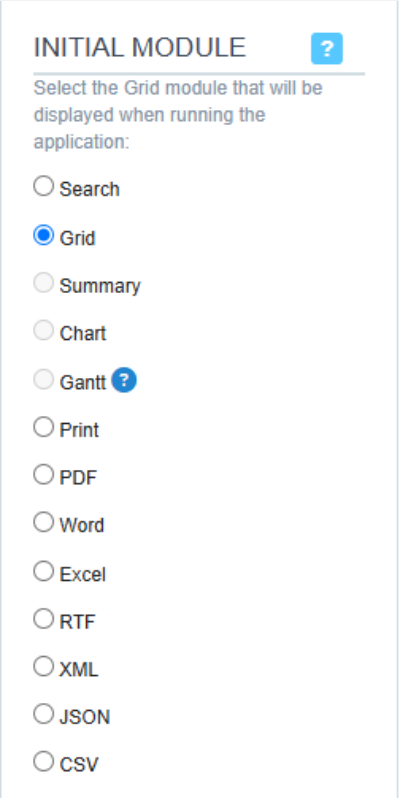
Button in the bottom left corner



Procedure Modules

Initial Module

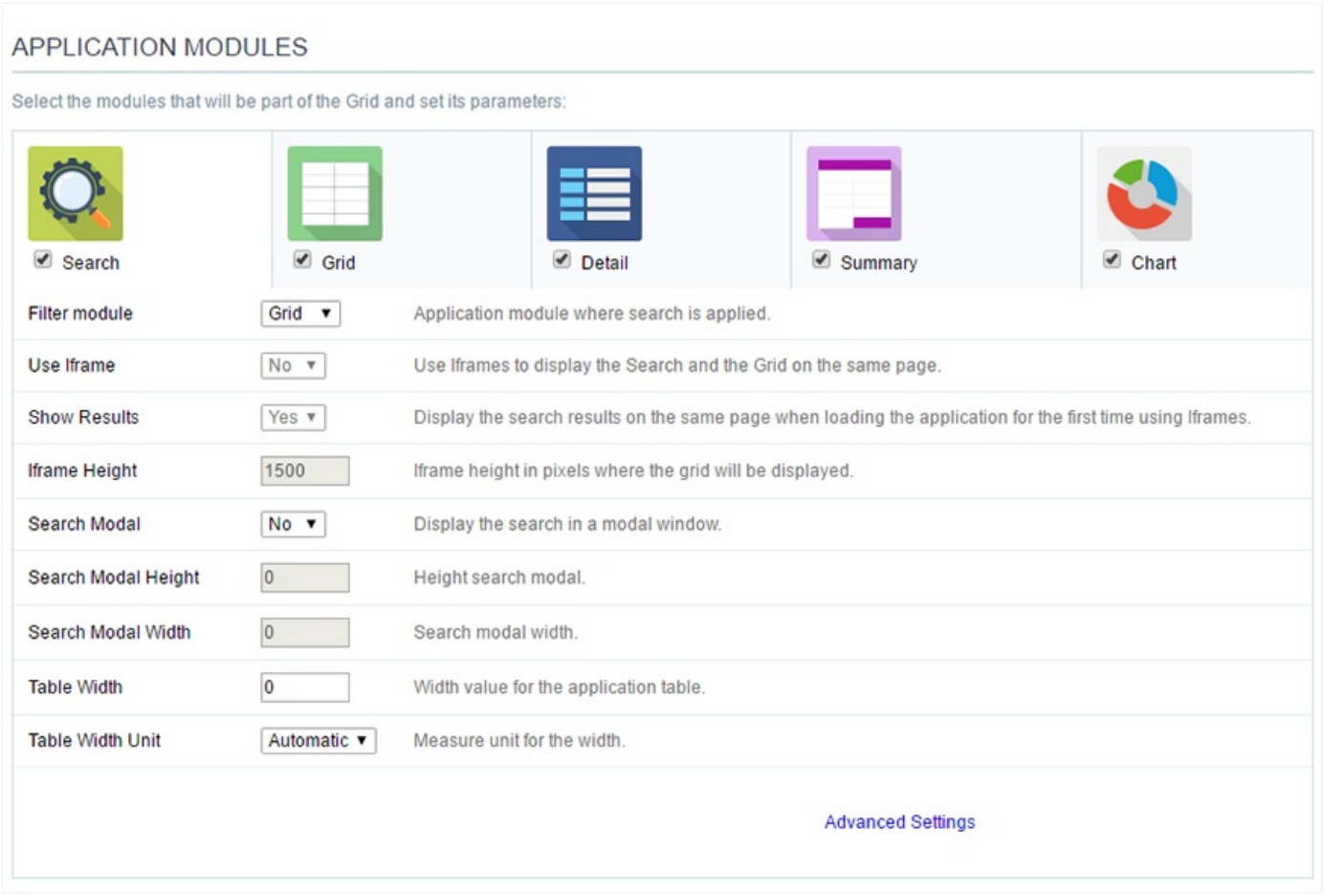
Allows you to set in which mode the application will start when executed.



Grid Application Modules

- **Filter** : Application will be started by the filter view, so you can filter the records before the next application, that can be a Grid, PDF, Summary... you can configure that options within the Filter's settings.
- **Grid** : Application will be started by the Grid itself, this is the default option.
- **Summary** : Application will be started by the Summary. To use the Summary as initial application it's mandatory to create at least one Group, using the Static Group By settings.
- **Chart** : Application will be started by the Chart. To use the Chart as initial application it's mandatory to create at least one Group, using the Static Group By settings.
- **Print** : Application will be started by the print mode, according to the select command.
- **PDF** : Application will be started with the option to view or download a PDF file.
- **Word** : Application will be started with the option to view or download a Word file(.doc or .docx).
- **Excel** : Application will be started with the option to view or download an Excel file (.xls or .xls).
- **RTF** : Application will be started with the option to view or download a RTF file (.tft).
- **XML** : Application will be started with the option to view or download a XML file.
- **CSV** : Application will be started with the option to view or download a CSV file.

Filter

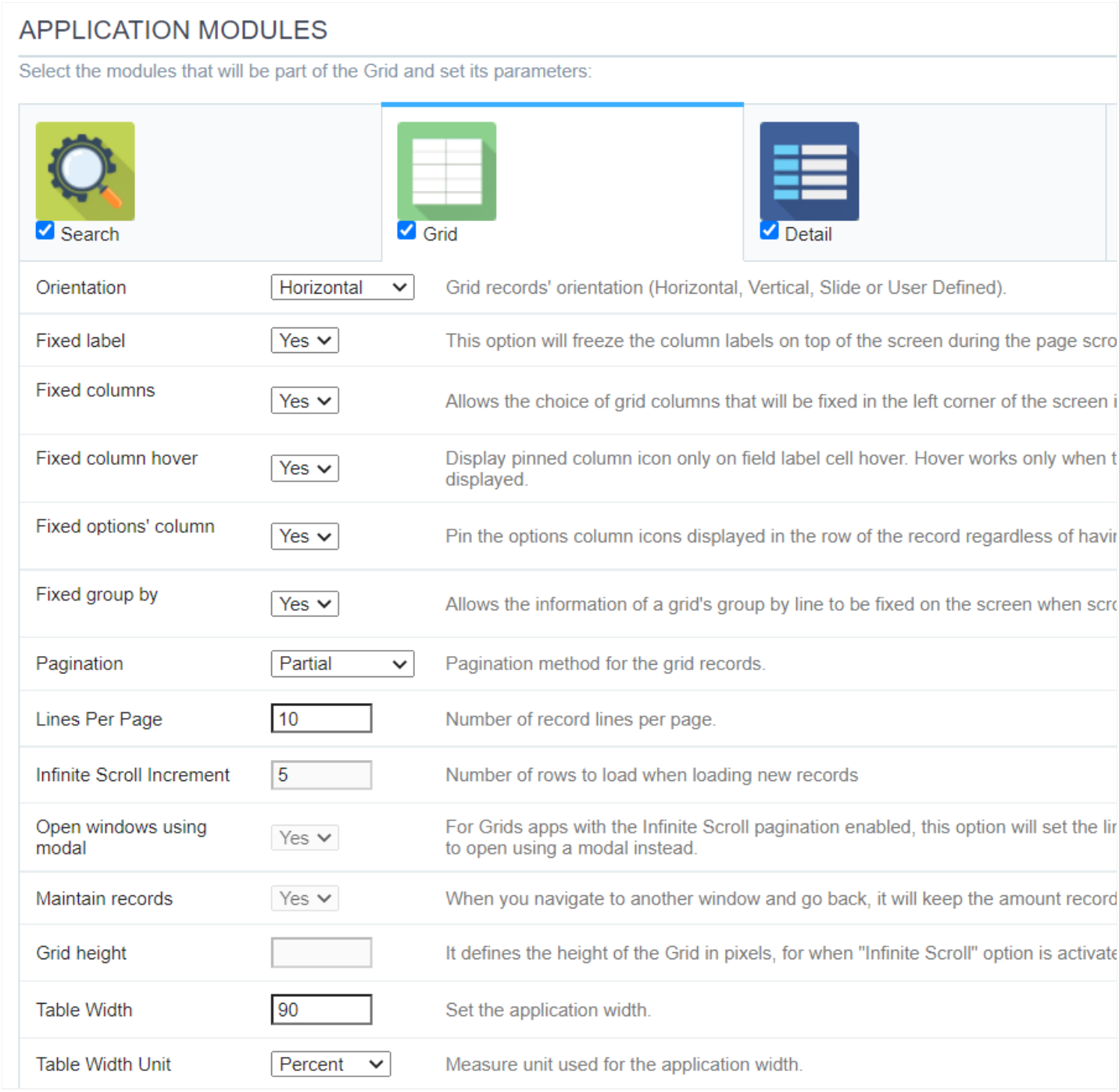


Grid

Filter Settings

- **Filter Module** : This option configures which application will be called after the Filter.
- **Use Iframe** : This option configures the filter to be displayed within an Iframe (displaying the Filter itself and the search results on the same page). This option is available only when the initial module is set to be the Filter.
- **Show Results** : Display the search results on the same page when loading the application for the first time when the “Use Iframe” option is set as “Yes”, otherwise, the results will be displayed only after the search.
- **Iframe Height** : Iframe height, in pixels, used to display the search results.
- **Search Modal** : This option configures the Filter Application to open in a modal window.
- **Search Modal Height** : Search Modal box height (in pixels). Option available only when the Search Modal box is enabled.
- **Search Modal Width** : Search Modal box Width (in pixels). Option available only when the Search Modal box is enabled
- **Table Width** : Filter application table width. This value can be in percent, pixel or automatic (set by Scriptcase). If you change the value you will also need to set the “Table Width Unit”
- **Table Width Unit** : This option configures the unit for the table width (percent, pixel or automatic). The value itself must to be set within the option “Table Width”.

Grid

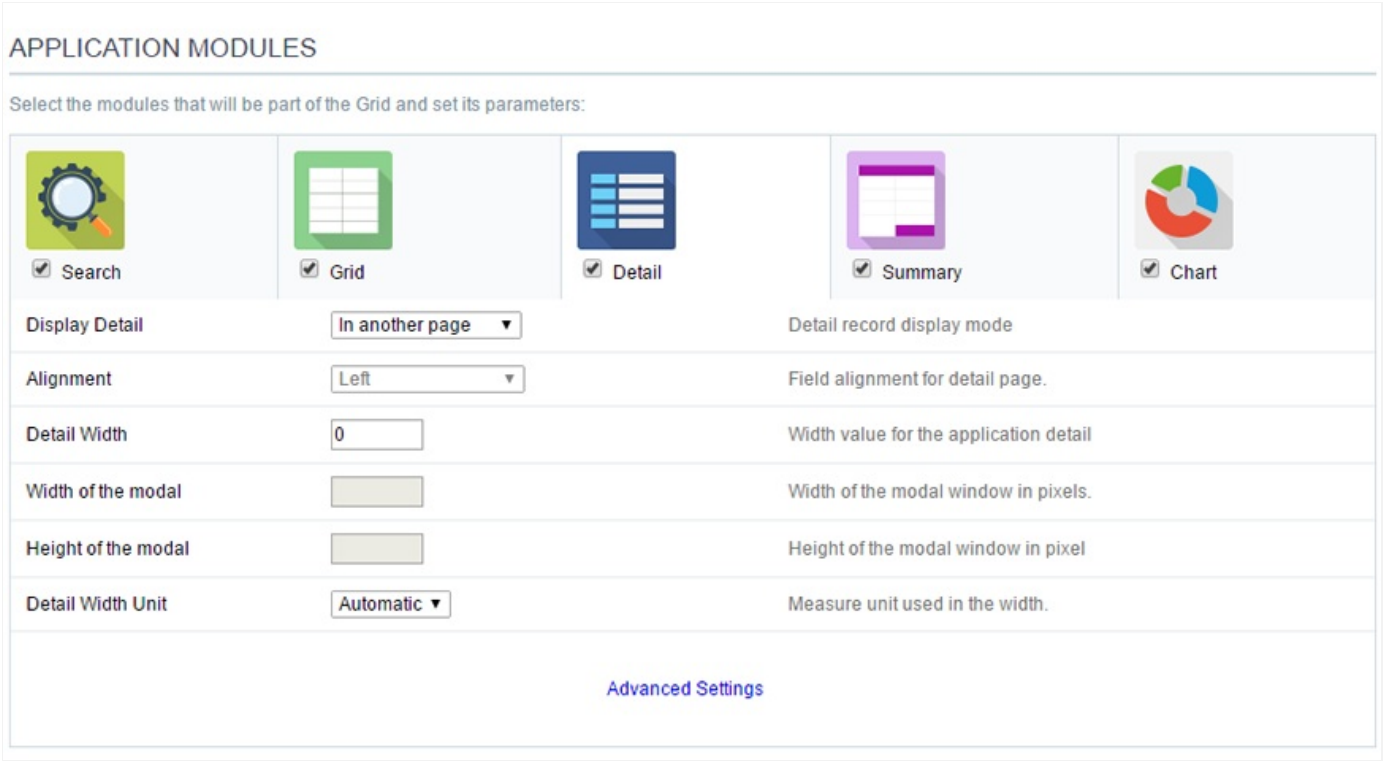


Grid Settings

- **Orientation** : This option configures the Grid records' orientation (Horizontal, Vertical, Slide or User Defined). When using the "User Defined" option you can design the HTML manually inside the option "Layout » HTML templates" and select within the Layout settings.
- **Fixed Label** : This option will fix the columns' label at the top of the page (it is only available when the Grud records' orientation is set as Horizontal).
- **Pagination** : This option sets the Grid paging type: Partial (pagination according to the amount of records set per page), Total (displays all records) or Infinite Scroll (automatic strolling according to the increment)
- **Lines Per Page** : This option sets the amount of records per page for the Grid
- **Infinite Scroll Increment** : This option sets the number of rows displayed on each increment of new records. It is available only when paging is configured with Infinite Scroll.
- **Open windows using modal** : For Grids Applications with the Infinite Scroll paging enabled, this option will set the links between applications and details of the Grid, that are configured to open in iframe, to open using a modal instead.
- **Maintain records** : It configures if the amount of records displayed will be preserved when the application navigates to another window and go back.
- **Grid height** : It sets the height of the query in pixels. If it is empty

- **Table Width** : Grid application table width. This value can be in percent, pixel or automatic (set by Scriptcase). If you change the value you will also need to set the “Table Width Unit”
- **Table Width Unit** : This option configures the unit for the Grid table width (percent, pixel or automatic). The value itself must to be set within the option “Table Width”.

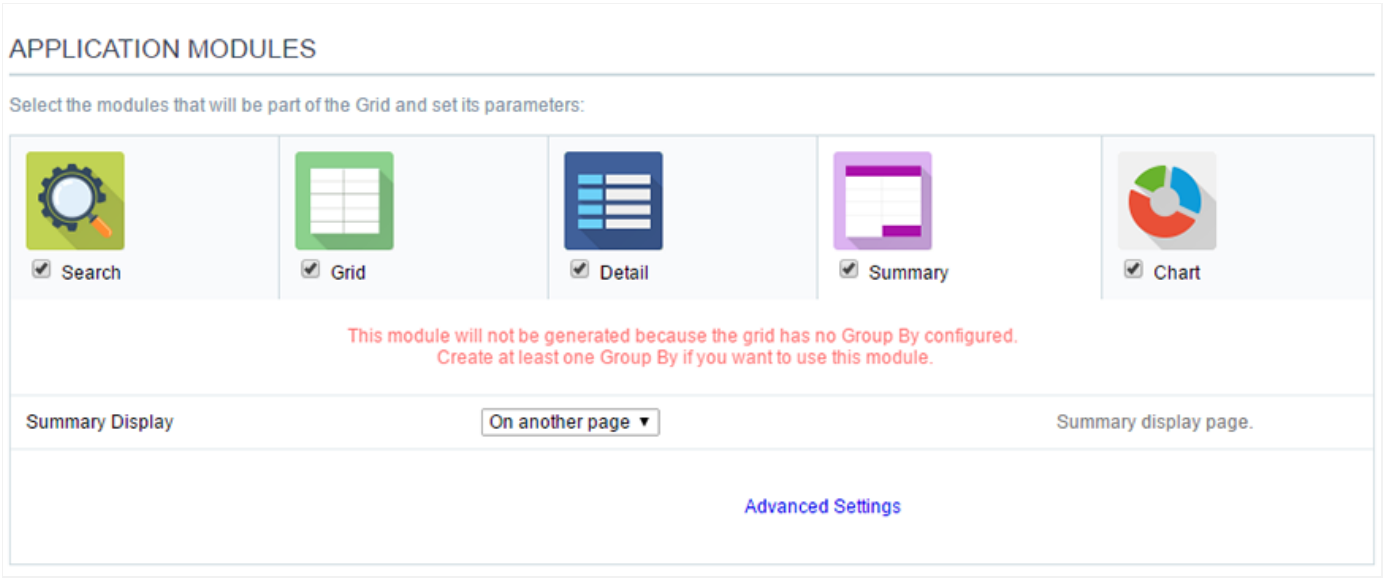
Detail



Grid Detail settings

- **Display Detail** : Allows you to set where the detail will open within the Grid Application * **Beside the Grid** : Displays the Grid Details to the right of the record, in the same window where Grid is being displayed. * **Below the Grid** : Displays the Grid Details below the Grid records, in the same window where Grid is being displayed. * **In another page** : Displays the Grid Details in another page, replacing the Grid view. * **In another window** : Displays the Grid Details in a separated browser window. * **Modal** : Opens a pop-up window to the display of the Grid Detail.
    - **Alignment** : Using this option you can set an alignment (center, right or left) for the Grid Detail when it’s using the “Display Detail” setting as “Beside the Grid” or “Below the Grid”.
    - **Detail Width** : This option sets the Grid Detail width when it’s using the “Display Detail” setting as “Beside the Grid”, “Below the Grid”, “In another page” or “In another window”.
    - **Width of the modal** : Allows you to customize the Grid Detail width when it’s using the “Display Detail” option as “Modal”.
    - **Height of the modal** : Allows you to customize the Grid Detail height when it’s using the “Display Detail” option as “Modal”.
    - **Detail Width Unit** : This option configures the unit for the “Detail Width” (percent, pixel or automatic).

Summary

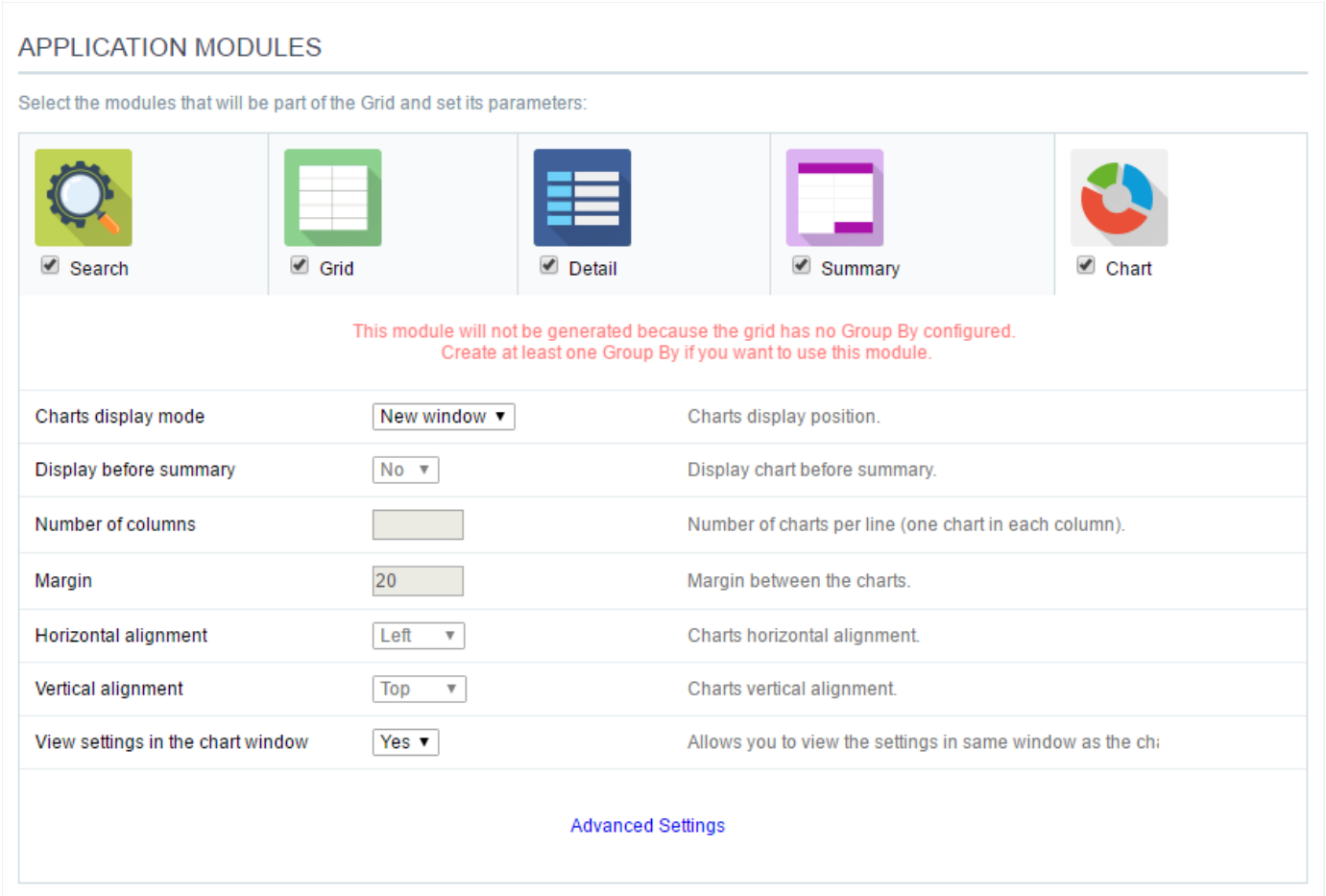


Grid

Summary Settings

- **Summary Display** : Sets the Grid Summary display option (On another page, On the last page, On every page). The Summary Application Module is only available when a group is created in the Group By settings.

Chart



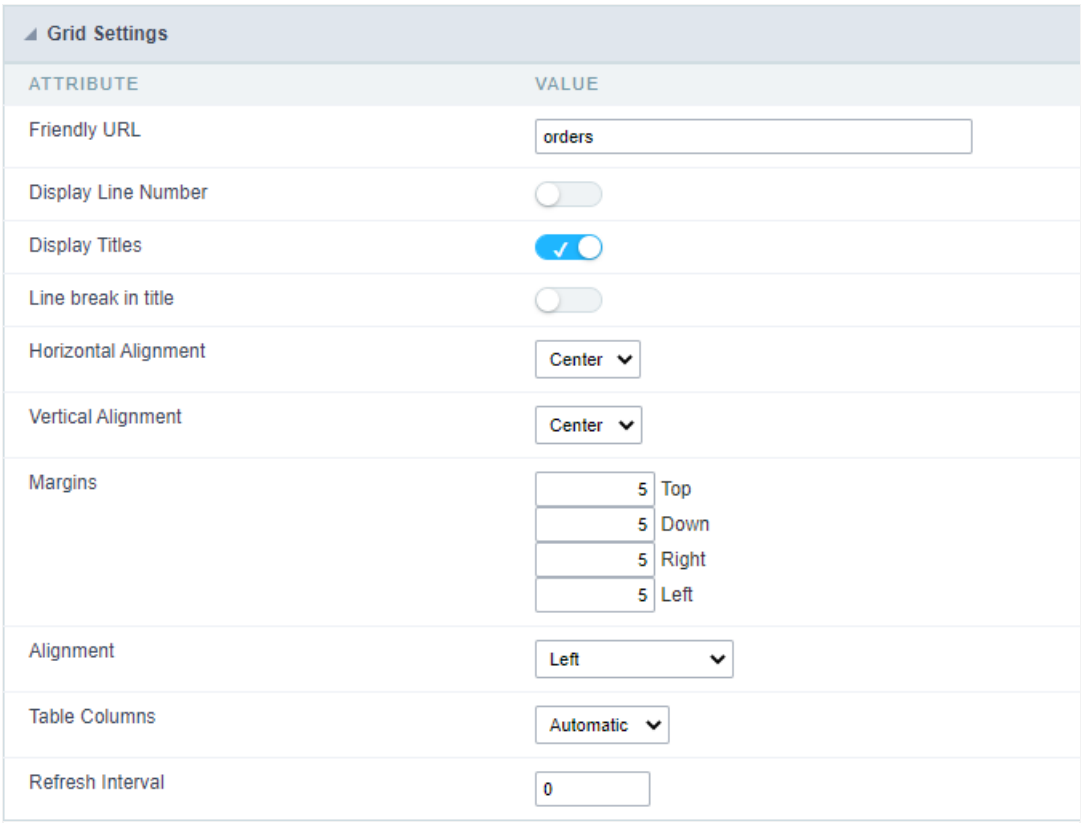
Grid

Chart settings

- **Charts display mode** : Sets the Chart Summary display option (New window, Same page, Other page).The Chart Application Module is only available when a group is created in the Group By settings.
- **Display before summary** : Sets whether the Chart is displayed above or below the Summary. Available when the Charts display mode is configured in the Same page.

- **Number of columns** : Sets the Chart number of columns per line (one in each column). Available when Charts display mode is configured in the Same page.
- **Margin** : Sets the Charts horizontal margin. Available when Charts display mode is configures in the Same page.
- **Horizontal alignment** : Sets the Charts horizontal position (Left, Right, Center). Available when Charts display mode is configures in the Same page.
- **Vertical alignment** : Sets the Charts vertical position (Top, Center, Bottom). Available when Charts display mode is configured in the Same page.
- **View settings in the chart window** : Sets the chart settings display on the same page. Available when the Char display mode is configured on Other page or new window.

Procedure Settings



Grid application advanced settings

- **Friendly URL** : This field allows you to change the URL that will be called by the application. Allowed characters are the same available on URLs: a-z, A-Z, 0-9, -_. This option can also be changed on the home screen, on the “Friendly URL” column at the applications list.
- **Display Line Number** : This option sets whether to display or not the sequence number of each Grid row.
- **Display Titles** : This option sets if the column title (column label) will be displayed or not.
- **Line break in title** : This option sets if the column line title will break or not.
- **Horizontal Alignment**: It allows to define the horizontal alignment of the application (Centered, Left, Right).
- **Vertical Alignment**: Allows you to define the initial vertical alignment of the application (Above, Centered and Below).
- **Margins** : Sets the application margins in pixels (up, down, right and left).
- **Alignment** : Allows you to set the fields alignment for when the Grid orientation is set as Vertical or Slide.
- **Table Columns** : Sets the column widths type: Provides (It will assume the informed width values at field level in the configuration of the visualization), Calculated (Calculate size according to type and the field size) and Automatic (The alignment will be according to the browser criteria).
- **Refresh Interval** : Allows you to set a reload interval for the page, in seconds. When is set as zero, there will be no page reload.

Procedure Edit Fields

This interface is useful for editing the field settings and their position to display.

Fields	Label	Datatype	Line Break	Line Break PDF	Title Horizontal Alignment	Text Alignment	Font Color
PAGE: PAG1							
BLOCK: GRID_ORDERS							
orderid	Orderid	Integer	☐	☐	Right	Right	
customerid	Customer Name	Text	☑	☐	Left	Left	
employeeid	Employeeid	Integer	☐	☐	Right	Right	
orderdate	Orderdate	Date	☐	☐	Center	Center	
requireddate	Requireddate	Date	☐	☐	Center	Center	
shippeddate	Shippeddate	Date	☐	☐	Center	Center	

Fields

Allows accessing the field settings (pencil icon on the left). You can change the field position by dragging them to the desired position. Drag a field to “fields not displayed” if you don’t want it in the app.

Label

Defines the title of a field in the app. For example: if the field name in the database is fld_txt_customer_name, you can display the label “Customer Name”.

Data Type

It informs the data type of the field.

Line Break

Allows the Line Break in the records when the field text is greater than the column width.

Line Break PDF

Allows the Line Break when the user exports the app as a PDF.

Title Horizontal Alignment

Defines the horizontal alignment of the field label.

Text Alignment

Defines the horizontal alignment of the field text.

Font Color

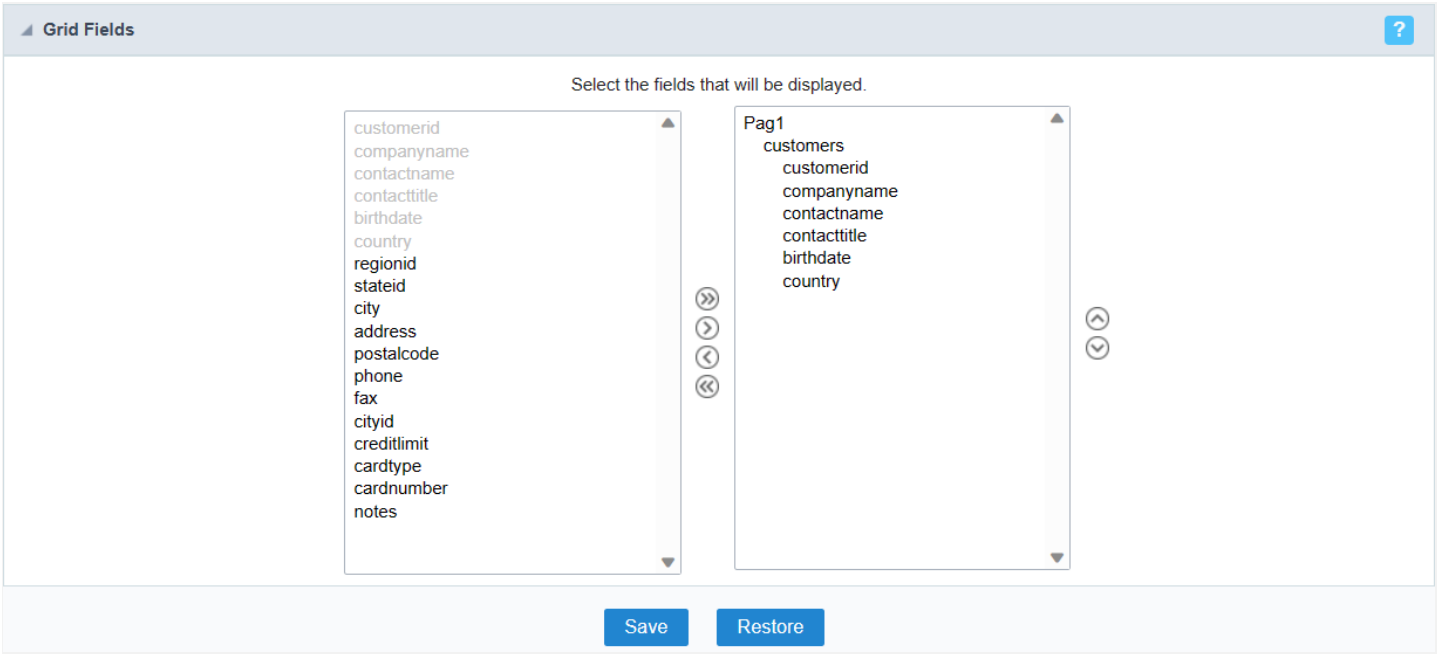
Configure the text color for the field.

Procedure Fields positioning

In this configuration screen we have a list of all the fields available in the application, whether they are fields mapped from the table or virtual fields (Created only in the Scriptcase interface).

It is also possible to allow the end user to manipulate the application's fields in the way they prefer, for that we must add the **columns button** in the application's toolbar.

In the **left column**, we have a list with all the fields available for use and in the **right column** we have a list of the fields that were selected to be displayed in the running application.



Adding and Removing Application Fields

The fields must be configured using the selection buttons, located between the columns.

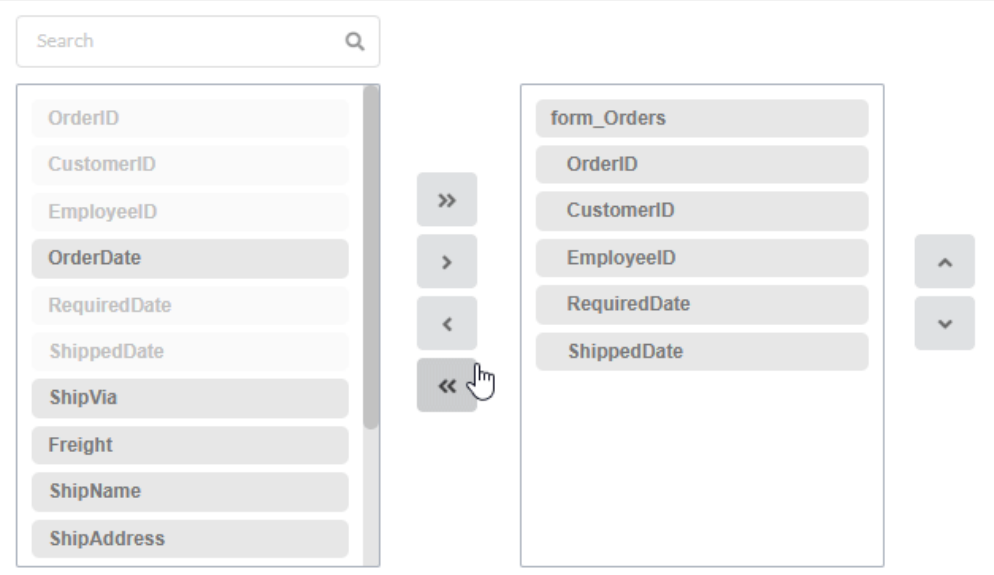
- » - Move all fields to the right.
- › - Moves only selected fields to the right
- ‹ - Moves only the selected fields to the left.
- « - Move all fields to the left.

In the left column, there are the application fields that are not in use in the filter, while the right column shows all the selected fields and the order in which they will be displayed.

To add fields to the filter, select the fields you want in the filter and click the › button.

To select fields in sequence, simply click on a field and drag. Pressing control (Ctrl) allows you to select multiple alternate fields.

Search and positioning



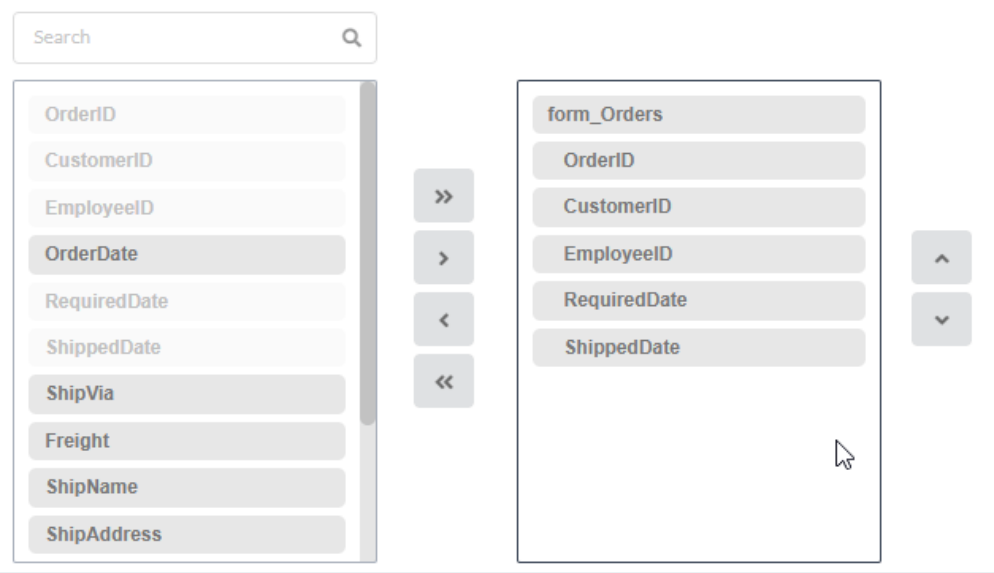
Field display order

The fields will be displayed in the application respecting the order defined in the right column.

The display order must be defined using the sort buttons and which will move the selected fields in the column.

Remembering that the fields must be positioned inside a block (**customers**) which must be inside a page (**Pag1**).

Check below the use of the buttons to order the fields

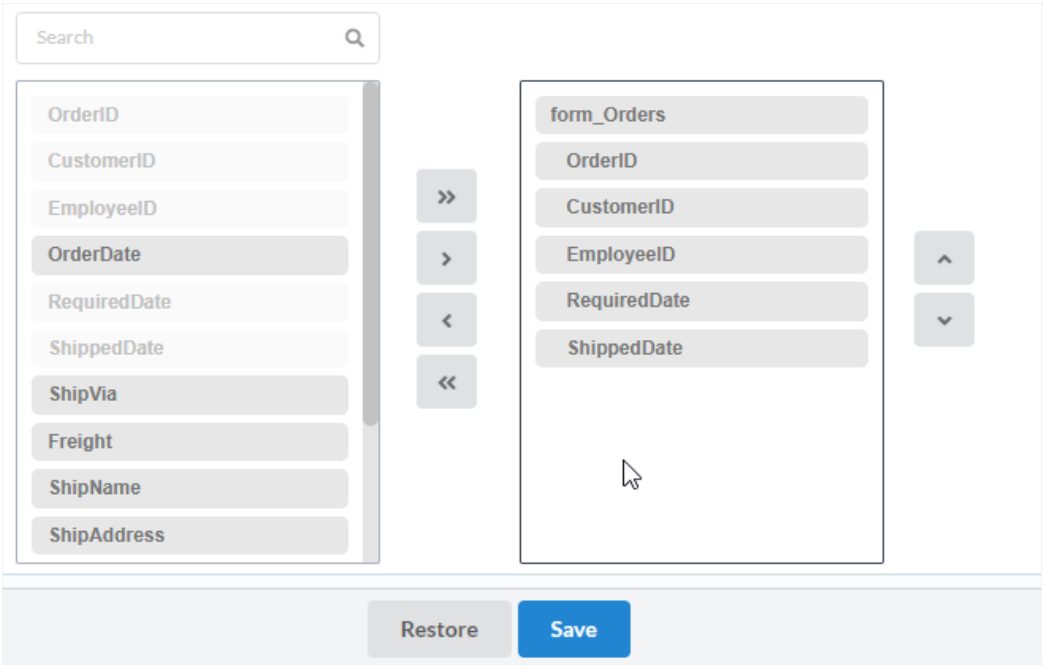


Save and Restore

Below the columns are two buttons:

- **Save** which records the current placement of fields in columns.
- **Restore** which reorganizes the fields returning to the last position saved.

Restoring the positioning of the fields to the last saved definition.

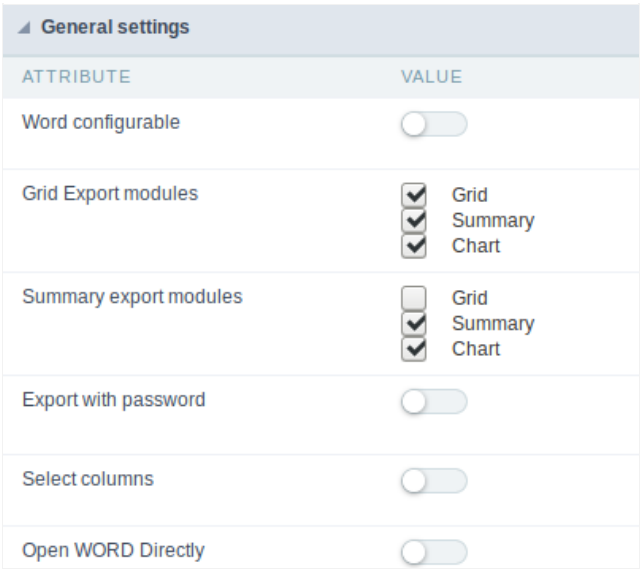

Search
OrderID
CustomerID
EmployeeID
OrderDate
RequiredDate
ShippedDate
ShipVia
Freight
ShipName
ShipAddress
form_Orders
OrderID
CustomerID
EmployeeID
RequiredDate
ShippedDate
Restore
Save

Procedure Toolbar

Procedure Export Settings

PDF Settings

WORD Settings



Word configurable

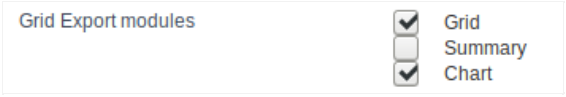
It allows the end-user to configure the export parameters on the fly.

Export Modules of Grid

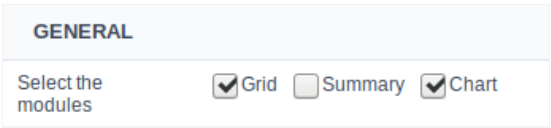
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

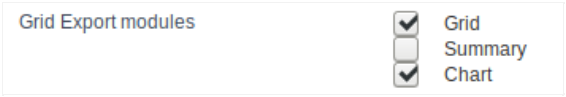


Summary Export Modules

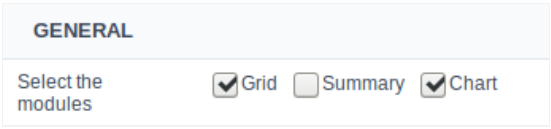
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.



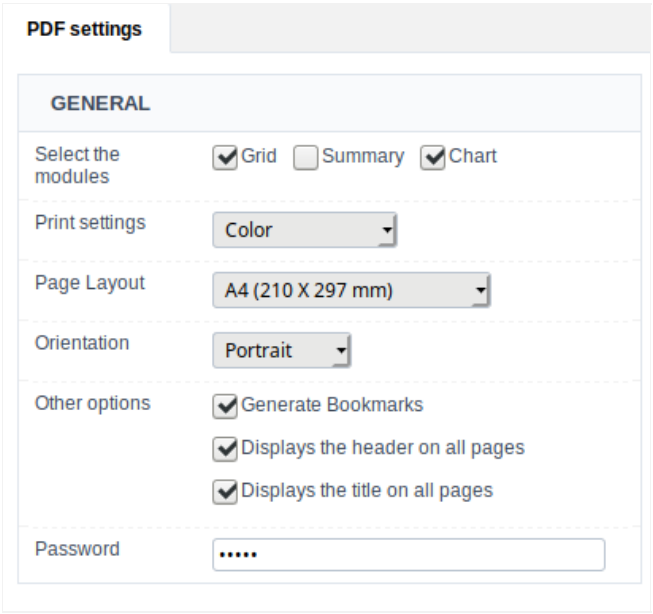
Export with password

This option allows the developer to set a password to export the file.

Password

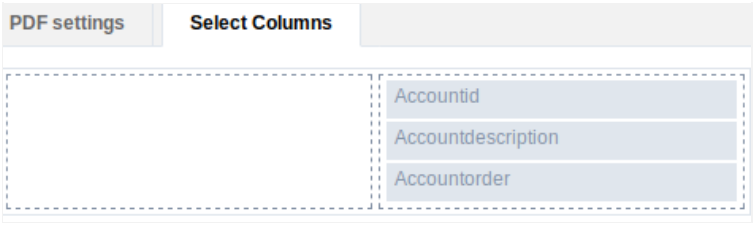
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

It allows the user has the option to choose the columns he wants to export at runtime.

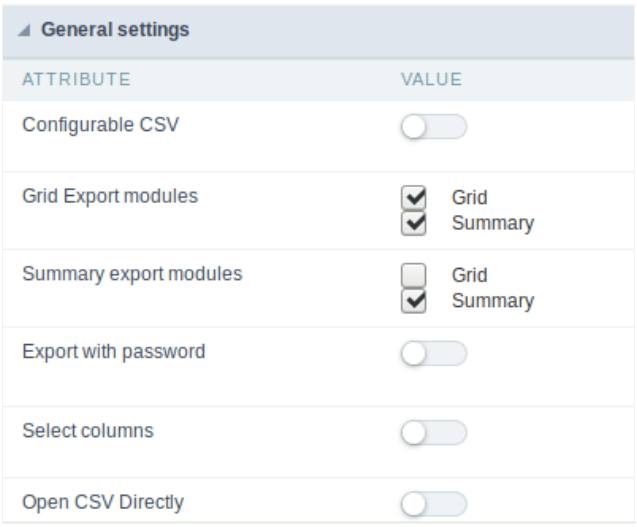


Open WORD Directly

Opens the generated WORD file without the need to display an intermediate page with a link to download it.

WORD Settings

CSV Settings



Configurable CSV

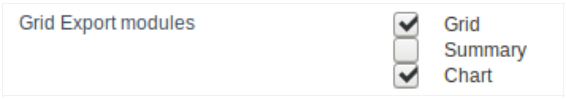
It allows the end-user to configure the CSV export parameters on the fly.

Export Modules of Grid

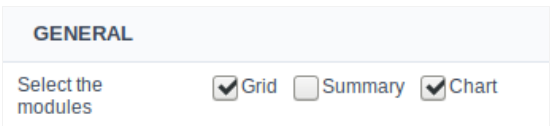
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

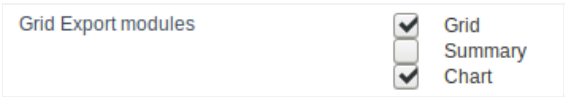


Summary Export Modules

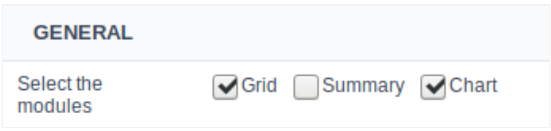
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.

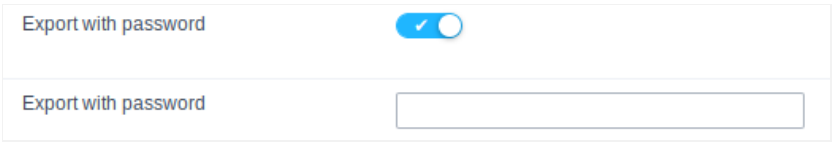


You can see the Summary option disabled at running time.



Export with password

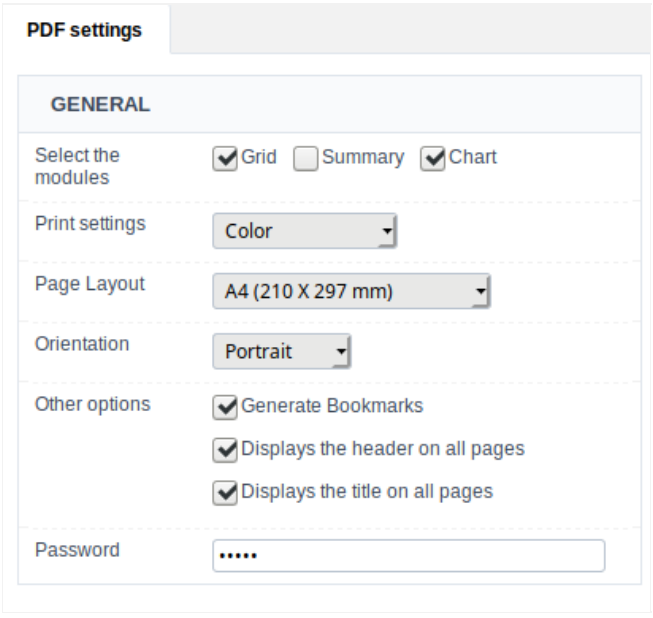
This option allows the developer to set a password to export the file.



Password

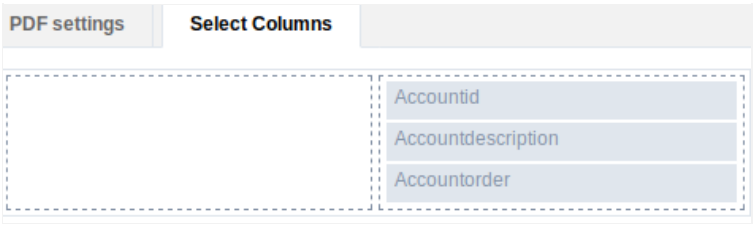
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

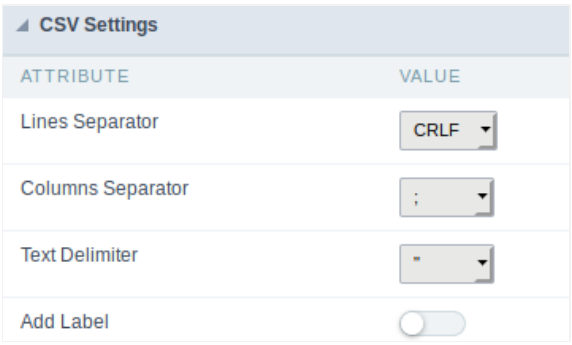
It allows the user has the option to choose the columns he wants to export at runtime.



Open CSV Directly

Opens the generated CSV file without the need to display an intermediate page with a link to download it.

CSV Settings



Line separator

It allows you to define the line separator character, each line representing a record.

Column separator

It allows you to define the column separator character, each column representing a database field.

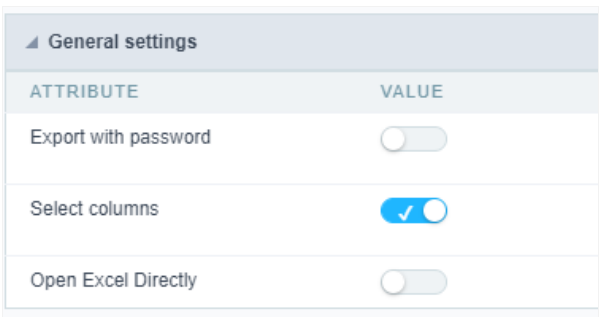
Text delimiter

It allows you to define the character used to delimit the text of the columns when we have database fields of type String.

Add label

It allows you to define displaying the label of the columns in the first line of the file.

XLS Settings



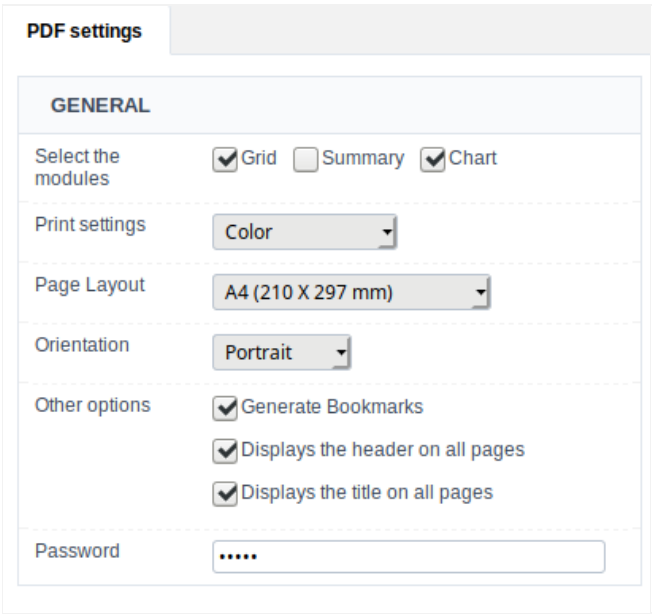
Export with password

This option allows the developer to set a password to export the file.

Password

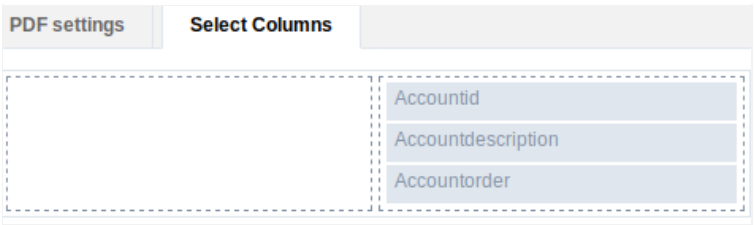
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

It allows the user has the option to choose the columns he wants to export at runtime.



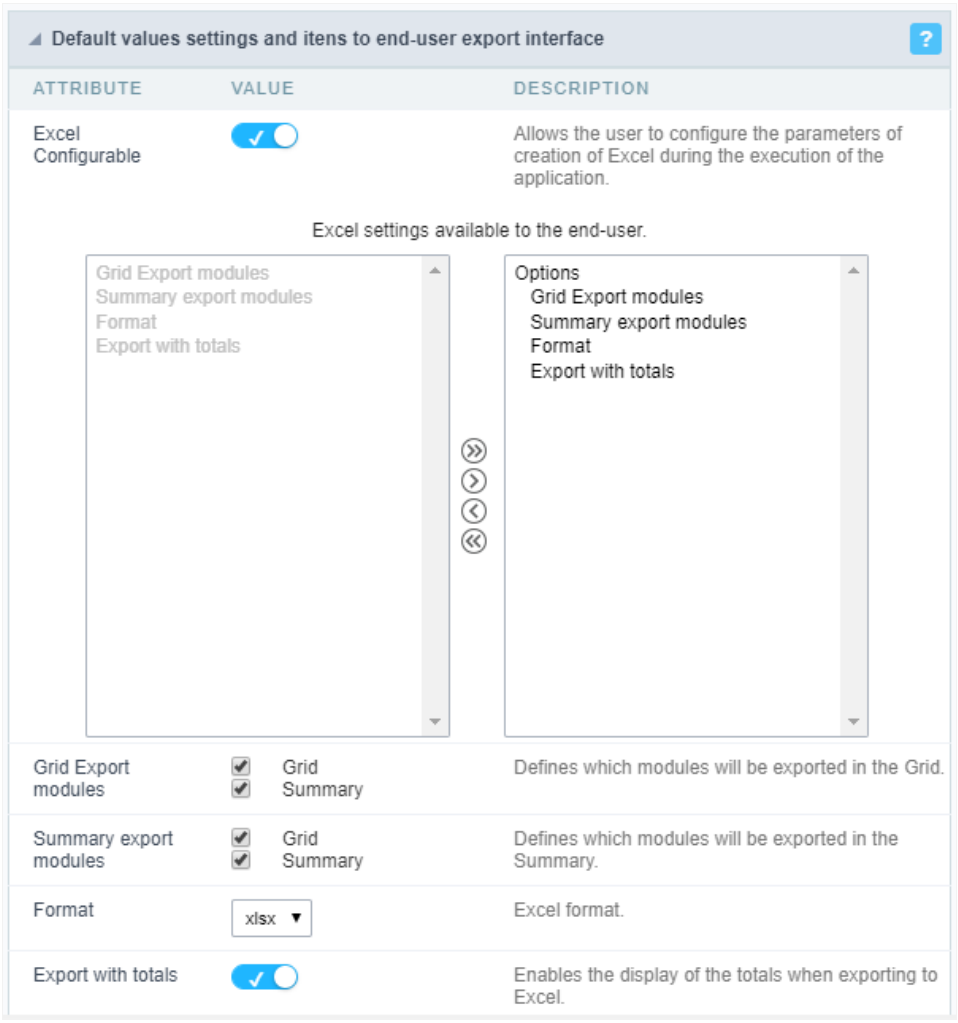
XLS Configurable

It allows the end-user to configure the XLS export parameters on the fly.

Open XLS Directly

It allows the end-user to configure the XLS export parameters on the fly.

XLS Settings

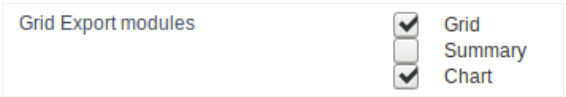


Export Modules of Grid

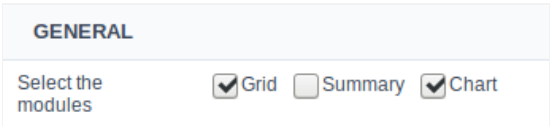
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

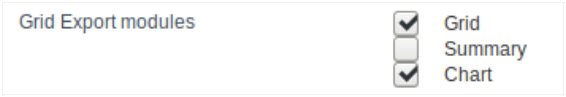


Summary Export Modules

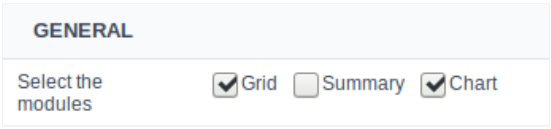
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.



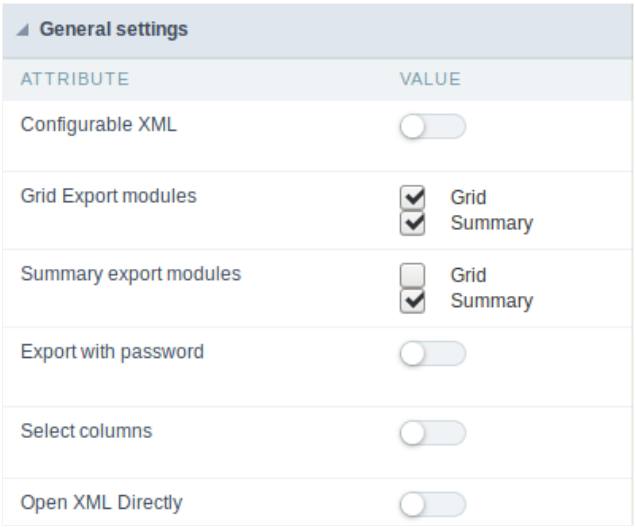
Format

It allows defining the Excel format of the generated document (xls or xlsx).

Export with totals

Show the totals when exporting the application data to Excel.

XML Settings



Configurable XML

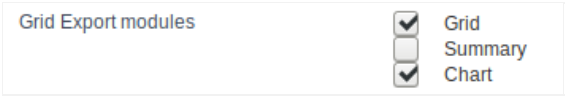
It allows the end-user to configure the XML export parameters on the fly.

Export Modules of Grid

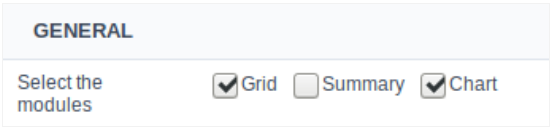
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

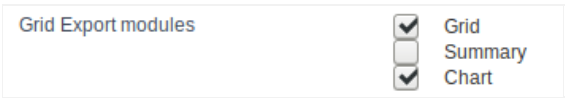


Summary Export Modules

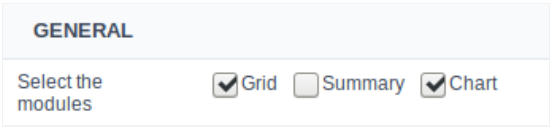
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.

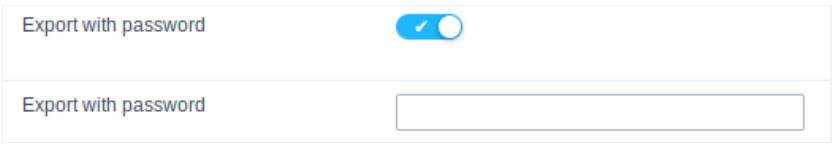


You can see the Summary option disabled at running time.



Export with password

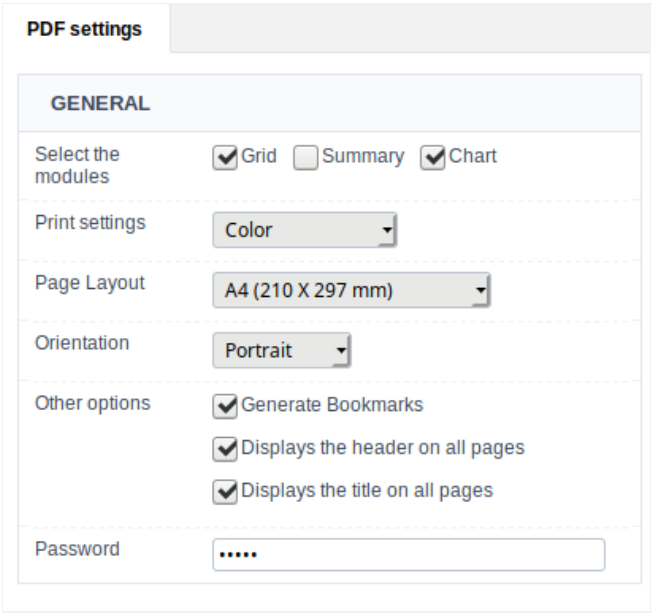
This option allows the developer to set a password to export the file.



Password

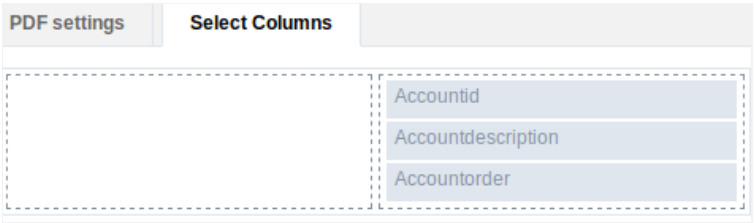
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

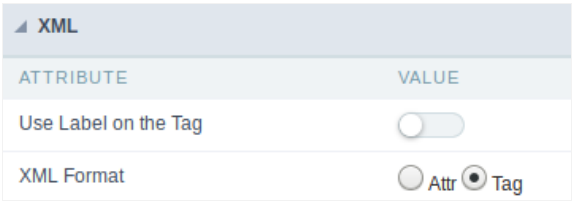
It allows the user has the option to choose the columns he wants to export at runtime.



Open XML Directly

Opens the generated XML file without the need to display an intermediate page with a link to download it.

XML Settings



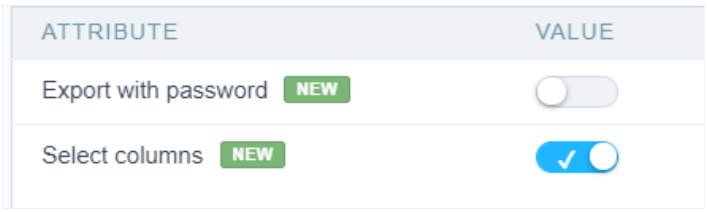
Use Label on the Tag

It allows using the field labels on Tags of the XML file.

XML Format

Define how to generate the records in the XML file, if it will store the values on attributes or new elements. (Attr or Tag)

JSON Settings



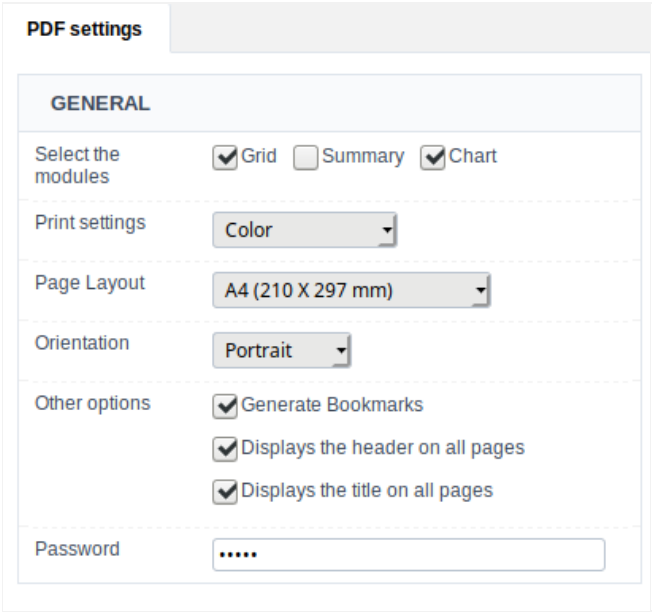
Export with password

This option allows the developer to set a password to export the file.

Password

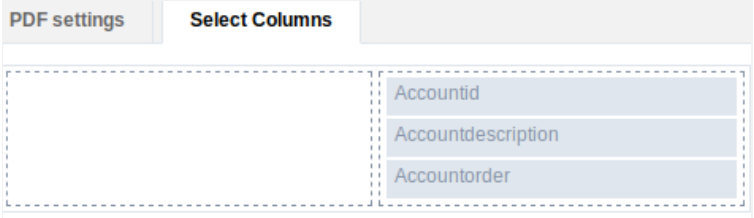
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

It allows the user has the option to choose the columns he wants to export at runtime.



Open JSON Directly

It allows the end-user to configure the JSON export parameters on the fly.

JSON Settings

Configurable JSON

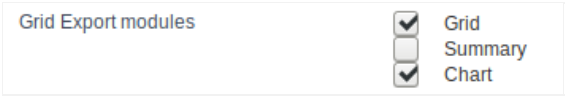
It allows the user to configure the XML creation parameters during the execution of the application. If this option is disabled, the end user will only have access to the settings defined by the developer.

Export Modules of Grid

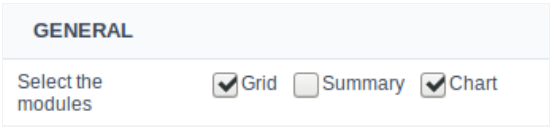
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

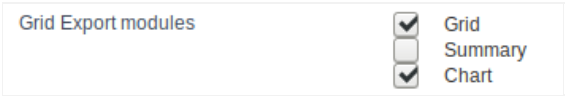


Summary Export Modules

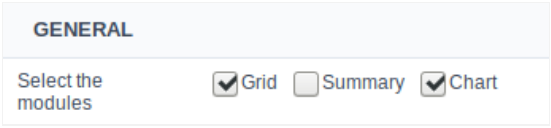
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.



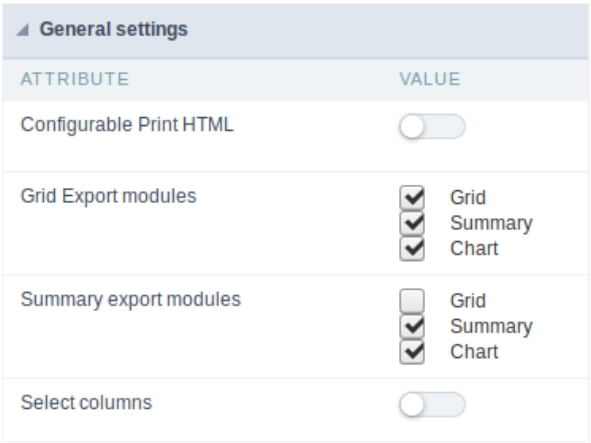
Use Label

It allows the label used in the application field to be sent in the JSON attribute.

Formatted value

It allows displaying the formatted numeric value or the value displayed in the database.

Print Settings



Configurable Print HTML

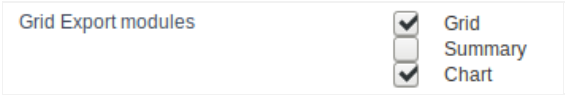
It allows the end-user to configure the HTML Printing parameters on the fly.

Export Modules of Grid

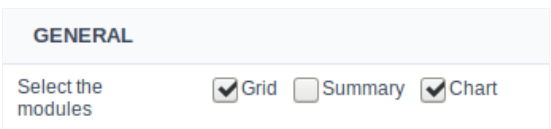
This option allows the developer to choose which modules are available in the grid export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.



You can see the Summary option disabled at running time.

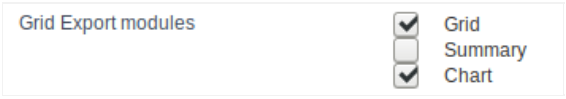


Summary Export Modules

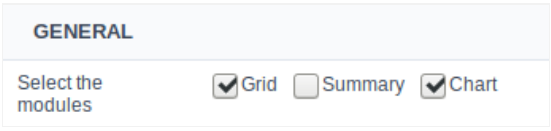
This option allows the developer to choose which modules are available in the summary export. The user can select the modules at runtime.

Example:

Disabling the Summary option in the development environment.

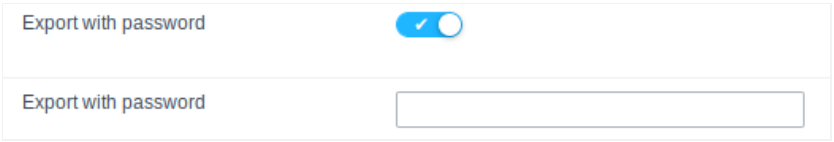


You can see the Summary option disabled at running time.



Export with password

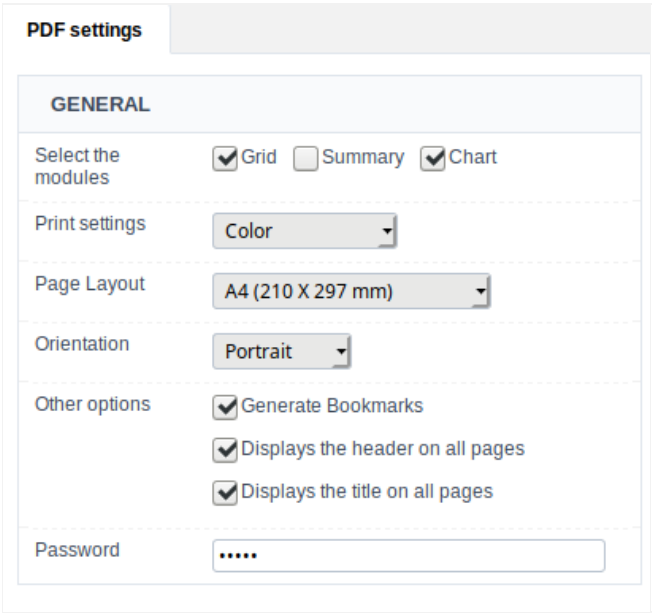
This option allows the developer to set a password to export the file.



Password

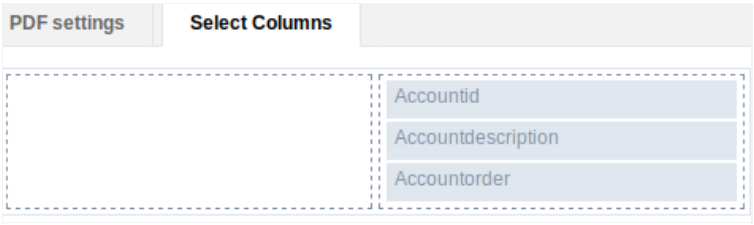
It allows the developer to set the requested password when exporting the file.

Runtime Example:



Select columns

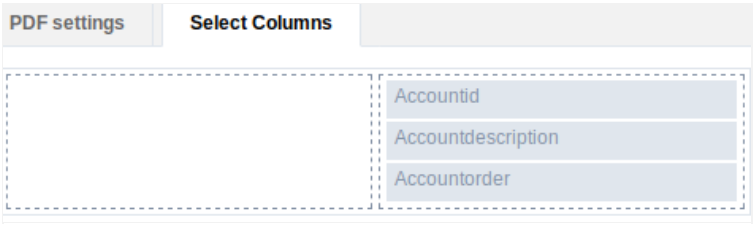
It allows the user has the option to choose the columns he wants to export at runtime.



RTF Settings

Select columns

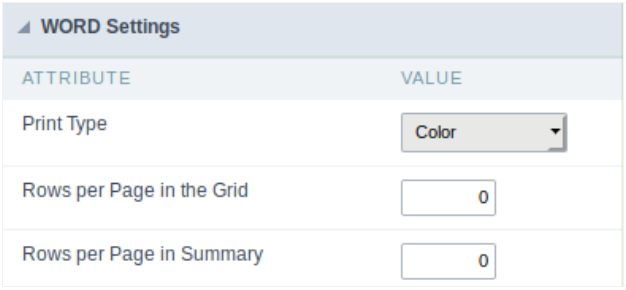
This option allows the end-user to select which fields he wants to export to the file.



Open RTF Directly

Opens the generated RTF file without the need to display an intermediate page with a link to download it.

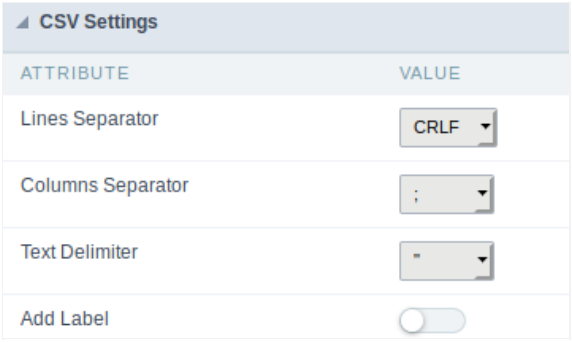
Word Settings



Word export settings

Print Type	Allows you to set the print mode for the word file (Both, Black & White, Color).
Rows per Page in Grid	Allows you to set the number of lines per page of the query will be displayed in the file.
Rows per Page in Summary	Allows you to set the number of lines per page in the Summary that be displayed in the file.
Open Word Directly	Allows you to set whether the document will be generated directly or if an intermediate page will be displayed.

CSV Settings



CSV export settings

Lines Separator Allows you to set the separator for character.

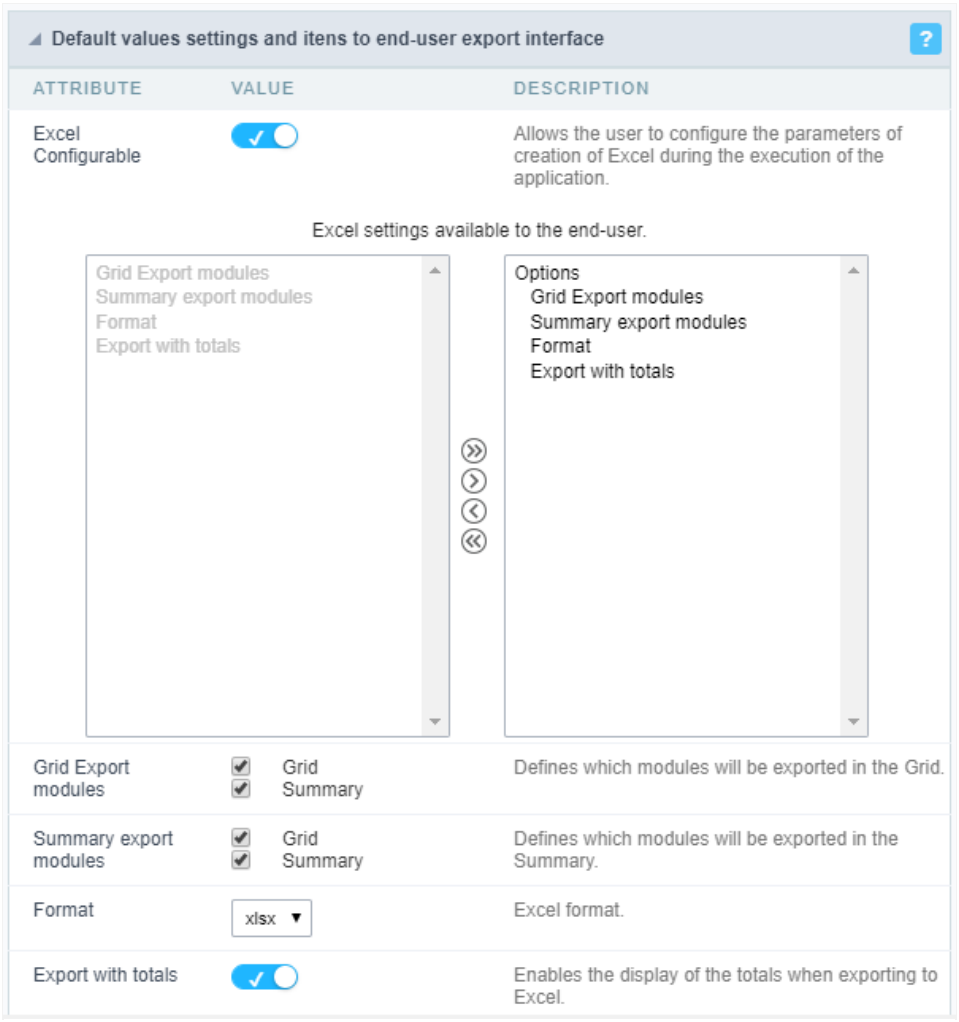
Columns Separator Allows you to set the column separator for character and records.

Text Delimiter Allows you to define the character used to delimit the text columns.

Open CSV Directly Allows you to set whether the document will be generated directly or if an intermediate page will be displayed.

Add Label Allows you to define if the columns label will be added to the file.

XLS Settings

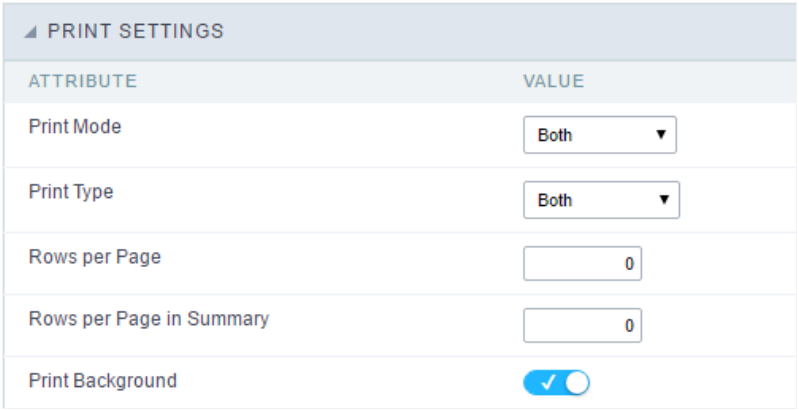


XLS export settings

Open XLS Directly Allows you to set whether the document will be generated directly or if an intermediate page will be displayed.

Format Allows you to define the format of the generated document (xls or xlsx).

Print Settings



Print export settings

Print Mode Allows you to define the contents of the file printing (Both, Current Page or Full Report).

Print Type Allows you to set the print mode for the file (Both, Black & White, Color).

Rows per Page Allows you to set the number of lines per page of the query will be displayed in the file.

Rows per Page in Summary Allows you to set the number of lines per page in the Summary that be displayed in the file.

Print Background Allows you to define whether the background will appear for printing.

Other Formats

XML and RTF export settings

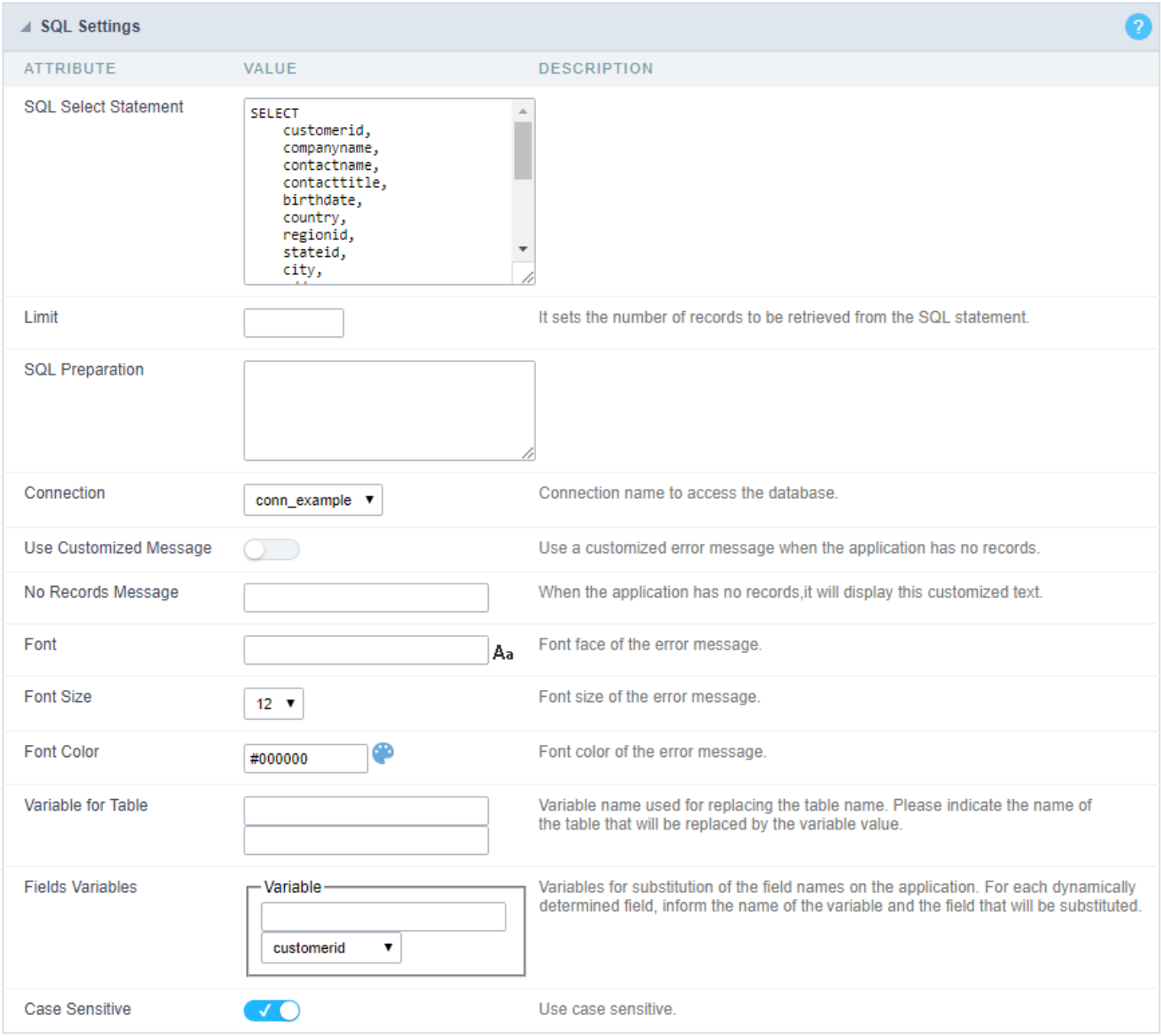
Generate XML directly Allows you to set whether the document will be generated directly or if an intermediate page will be displayed.

Generate RTF directly Allows you to set whether the document will be generated directly or if an intermediate page will be displayed.

Procedure SQL Settings

SQL Settings

This interface allows configuring the related database settings, such as the SQL statement, the used database connection, case sensitive, and others.



Grid SQL configuration

SQL Select Statement

It allows you to define the primary SQL of the application. You can edit this SQL to add or delete fields.

Limit

Lets you limit the display in the number of records retrieved by SQL query.

SQL Preparation

You can enter SQL commands or procedure names to execute them before the primary SQL of the application.

Connection

It allows defining the database connection of the application. You can change the connection to another one that has the same table.

Use Customized Message

Lets you define to display the “no records” message or not.

No Records Message

Lets you set the message when the application has no records.

Font

This option is available when using the “Use Customized Message” option. It lets you set the font for the message.

Font Size

This option is available when using the “Use Customized Message” option. It lets you to set the font size.

Font Color

This option is available when using the “Use Customized Message” option. It lets you to set the font color.

Variable for Table

It allows to use a variable to change a part of the string containing the table name.

Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the table you want to replace (replace from).

Fields Variables

It allows to use a variable to change a part of the string containing the field name.

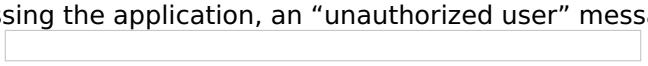
Fill the first input with the name of the variable (replace to). The second input you should fill with the part of the name of the field you want to replace (replace from).

Case sensitive

It defines if the database connection uses case sensitive or not.

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application. 

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Headers

Desabilitar o Auditor XSS

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

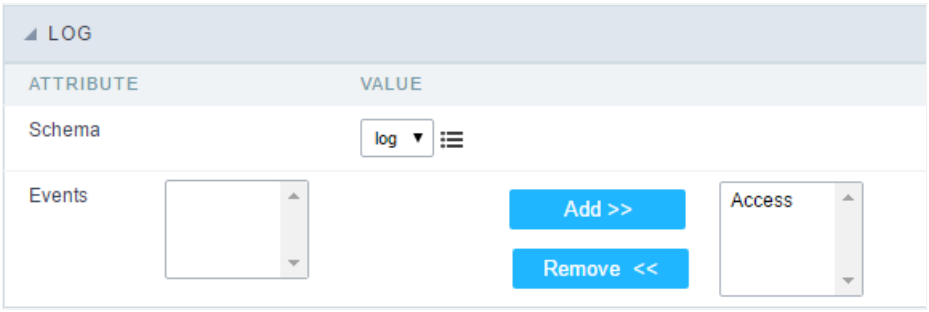
Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

Navigation Policy Example `navigate-to example.com`

Procedure Log Configuration

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Procedure - Fields Overview

The applications can have two types of fields.

Table Fields

Columns of the table connected to the application and their data types.

Example:

In a table with the following structure:

```
CREATE TABLE `orders`  (
  `orderid` int(11) NOT NULL,
  `customerid` varchar(5) NULL DEFAULT NULL,
  `employeeid` int(11) NULL DEFAULT NULL,
  `orderdate` date NULL DEFAULT NULL,
  `requireddate` date NULL DEFAULT NULL,
  PRIMARY KEY (`orderid`) USING BTREE
);
```

These are the listed fields on left menu **Fields**:

Virtual Fields

Fields created inside application to assist in the development process.

How to create a virtual field

1. Inside an application, access the left menu **Fields** and click on **New Field** option.
2. Choose how many fields do you want to create:

1. Create the fields and define type, name and label for each one:

- **Type**: Data type of the field.
- **Name**: Internal name of the field. Used to identify the field on events and Scriptcase interface.
- **Label**: Title of the field displayed on the application.

  Scriptcase **doesn't insert** virtual fields on the table after its creation.

Types of fields by applications

Application	Table Fields	Virtual Fields
Grid		
Procedure		
Chart		
Form		
Control		
Calendar		
Search		
PDF Report		

The applications **Menu**, **Tree Menu**, **Dashboard** and **Blank** were not listed because they haven't manage fields.

Procedure - Text Field

General Settings

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the "Separated by" field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.

- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).

- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).

- **Background Image** : Allows you to add a background image to the field.
- **Line Break** : Enables the possibility of a line break for the field.
- **Content color** : Changes the content color of the query field and the grouped field, when available.
- **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

//## PDF Configuration {#id-04}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - Multiple Lines Text Field

General Settings

Multiple Lines Text field Configuration Interface.

- **Data Type** : Define the type of field for the application. When it is defined as a Multiple Lines Text, it accepts letters, numbers and special characters in multiple lines.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be more clear for the comprehension of the user if the name would be “Client Name”.
- **Case Settings** : Convert the letter from the field when losing focus. The options are:
  - **Upper case** : All in Upper Case
  - **Lower case** : All in Lower case
  - **Capitalize first word** : Capitalizes the first letter of the first word
  - **Capitalize all words** : Capitalizes the first letter of all the the words
- **Show HTML content** : Determines if the HTML contained in the field will be displayed or not. If enabled, the HTML will be displayed, otherwise the HTML will be interpreted by the browser.
- **Grid Mask** : Defines the mask for the field display.There are two typed of masks described below:

Character	Description
X	Placeholder to any character. Replaced by any character. If number of characters entered are less then the mask size, the field value is completed with zeros (Filling full size field entry is required).
Z	Replaced by any character retrieved from database. Suppress zeros at field left (Complete field filling is optional). When used combined with the mask character X it should be placed at the mask left.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

- **Do Not Repeat Value** : Do not repeat the value of the field in case it is the same as the previous record.

- **Run content in JavaScript** : If enabled, the JavaScript will be interpreted by the browser, otherwise the JavaScript will be displayed.
- **SQL Type** : Informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or

bottom).
- **Background Image** : Allows you to add a background image to the field.
- **Line Break** : Enables the possibility of a line break for the field.
- **Content color** : Changes the content color of the query field and the grouped field, when available.
- **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

//## PDF Configuration {#id-03}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - Integer Field

On this page, you will learn how you can configure settings related to the Number field. From the use of specific symbols display to the mode in which they are displayed. And thus, boost the application.

Data type:

It sets the application field type. When the field is set as Number, it is permissible to define formatting rules of integers.

Label:

It sets the title that will be displayed in the field when you run the application. The terminology used in the interface is fundamental for your system has a good usability. In this case, we recommend you to use names and terms familiar to the end user of the application, instead of using terms originated in the system.

Field Mask Input:

It sets the mask input for the field. There are two types of mask as described in the table below:

Character	Description
X	It is overridden by any characters returned by the database. Filled is required and the value will be completed with leading zeros when there are fewer characters than the size of the mask.
Z	It is overridden by any characters returned by the database. Its filling is optional and when there are fewer characters than the mask size, nothing will be done about the characters that are missing. In addition the leading zeros will be suppressed. When used in conjunction with the X necessarily the use of this character is to the left of the mask.
9	It represents any numeric character (from 0-9)
A	It represents an alpha numeric character (A-Z,a-z)
*	It represents any alpha-numeric character (A-Z,a-z,0-9)entered by the user.

Mask Input Examples:

Field	Mask Input	Typed Value	Formatted Value
Phone Number	+99 99 9999 - 9999	123456789012	+12 34 5678 - 9012
Phone Number	(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
Phone Number	(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
Phone Number	(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
Software Key (Letters only)	AAAA-AAAA-AAAA-AAAA	QWERTYUIASDFGHJK	QWER-TYUI-ASDF-GHJK
Software Key (Letters and numbers)	xxxx-xxxx-xxxx-xxxx	Q1W2E3R4T5Y6U7I8	Q1W2-E3R4-T5Y6-U7I8
License Plate	AAA - 9999	QWE1234	QWE - 1234
Scriptcase Serial Key	A999A999A999-**	D111H111G111DG2P	D111H111G111-DG2P
Multiple masks (Phone Number)	9999-9999;(99)9999-9999;	9999 999 9999	+99 99 9999-9999

Repeat value:

This option when enabled will allows you to repeat the field value if it is equal to the value of the previous record in the database.

Example:

Tipo SQL:

Reports the SQL type of the field as it is configured in the database.

Use regional settings:

It allows you to apply regional settings to the number fields formating. When this option is not selected it will be displayed the attributes group, minus sign and negative number format.

Grouping:

It allows you to define which character is used to separate thousands. This option is only available when the option to use regional settings is disabled.

Negative sign:

It allows you to define which character will be used for the display of negative numbers. This option is only available when the option to use regional settings is disabled.

Negative number format:

It allows you to define the placement of the negative sign in relation to value. This option is only available when the option to use regional settings is disabled.

Color for the negative values:

It allows you to define a color when the value is negative, improving the understanding of the end user about that kind of value.

Example:

Display the value in words:

The value of the field will be displayed in full on application. This feature can facilitate the comprehension and understanding of the user.

Example:

Line size:

Maximum size in characters to be displayed in the value cell, in full. When this value is exceeded the line will break within the cell.

Lookup Settings

The Lookup is a way used to provide the end user a list of values description/name where the items can be selected and used in the application. For example, for a field sex that in the database is represented with M or F values can be presented as male or female dynamically with the lookup. So you will be able to retrieve these values from a database table or manually, if the reference table does not exist.

Example:

Lookup Methods

There are two lookup methods available:

Manual: To use this method it is necessary to inform the selection conditions manually to the Lookup.

Automatic: To use this method it is necessary to use a lookup query from the database to access the reference table.

Editing-automatic lookup:

Select Command:

It sets the SQL command that will retrieve the value to display inside the Grid field. The command must have the following format

```
SELECT field_to_be_diplayed FROM table WHERE key_field = {grid_field}
```

> The Grid field must be always referenced between curly braces {}, So at run time, the field braces will be replaced by its value in the table.

Multiple Options:

When the Select command informed return multiple values, the option must be selected.

Delimiter:

It sets the tab of values returned by the Select command, this option should be filled when the “Yes” option is checked in the multiple attribute options.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Use a specific connection:

This option allows you to select another existing connection in the project, other than the current connection, to select the values for the lookup.

Edit Lookup - Manual:

Lookup used to modify the display of the field with predefined values.

Lookup Types

It sets the operating mode of the lookup can be: single value, multiple values (delimiter), multiple values (position) and multiple values (binary).

Single value: it will be displayed only one value for the field. There is a direct replacement of the value stored in the Bank for a label defined in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value of “M” will be replaced by “Male”.

Multiple Values (Delimiter):

By using this option it will be displayed multiple values for the selected field. The values must be separated by a delimiter to be informed. A replacement of parts from a string stored in the field, separated by a delimiter for values contained in a list.

Label: Text that will be displayed in the Grid.

Value: Attribute that matches the value stored in the table. For example, the value S;M will be replaced by Sports and Movies.

Delimiter: Character used to separate the values within the same string.

Multiple Values (Position)

Allows you to recover information stored in a single string of the selected field. In order for this information to be retrieved must be informed, in addition to the label, the starting position and the number of bytes that each information occupies within the string.

As an example we will use a string to store Sex, Marital status and Hobby respectively. Sex occupies one byte, Status one byte and Hobby two bytes .

For this purpose we define the list as:

Label	Value	Start	Size
Male	M	1	1
Female	F	1	1
Married	M	2	1
Single	S	2	1
Study	ST	3	2
Sportes	SP	3	2
Reading	LE	3	2

Example: the string MMST would be presented in the query as: Male Married Study

Label: Text that will be displayed in the Grid.
Value: Attribute that matches the value stored in the table. For example, the value M will be replaced by Male.
Start: : Starting position in the string where the information is recorded. The first position is always 1.
Size : Number of bytes that the information occupies in the string.

Multiple values (binary):

It allows you to retrieve several information stored in decimal form of the selected field.

As examples we will use the following list (although not informed in the interface of inclusion of values in the list, each item has a value assigned automatically according to the order):

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

To display the data, a decomposition of the decimal number stored in the bank is performed. For example, numbers 11 and 12 (written in the database) would be decomposed as follows:

11 = 1 + 2 + 8 = (Sports - Culture - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

12 = 4 + 8 = (Leisure - Reading)

Assigned value	Description in Lookup
1	Sports
2	Culture
4	Leisure
8	Reading
16	Music

Label:

Text that will be displayed in the Grid.

Include Button:

Includes in the list the values populated in the Label and value fields.

Alter Button:

It changes the attributes of the selected item.

Delete Button:

Deletes the selected item.

Clean Button:

It cleans the fields

Save Button:

Allows the user to save the entire list, for later use using the Load Lookup definition option.

Load Lookup definition:

Allows you to load a predefined value list for use as a lookup.

Use the lookup in:

Grid: It applies the lookup in all Grid modules (HTML, PDF, XML, CSV, and RTF).

Summary: It applies the lookup only to the Summary (HTML and PDF).

Both This is equivalent to check Grid and Summary options.

Default Value:

Defines a default value, used in cases where the stored value does not match any value from the list.

Display original value and lookup:

When selected **No**, is only displayed the value returned by the Select command. Otherwise, it returns the original value of the field and the value that is returned by the Select command separated by character set in separated by.

Separated By:

Sets the character(s) that will be used to separate the original value of the field and the value that is returned by the Select command. This option should be filled when is checked **Yes** in the Show original value and lookup.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:
- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

  In the example shown above, the width of the Chart is 200 pixels.
- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.
- **Display Value** : It will display the value and the chart at the same time, the value being on the left or

right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

//## PDF Configuration {#id-06}

Charts Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - Decimal Field

General Settings

Decimal field Configuration Interface.

Data Type

Define the field type to Decimal. It allows defining the Decimal number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values format

 Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the

select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.

- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.

- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:

- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

  In the example shown above, the width of the Chart is 200 pixels.

- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.

- **Display Value** : It will display the value and the chart at the same time, the value being on the left or right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this

field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

//## PDF Configuration {#id-06}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - Percent Field

General Settings

Percentage field Configuration Interface.

Data Type

Define the field type to Percentage.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value

returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and

lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:
- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

  In the example shown above, the width of the Chart is 200 pixels.

- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.

- **Display Value** : It will display the value and the chart at the same time, the value being on the left or right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not

informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

//## PDF Configuration {#id-06}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - Currency Field

General Settings

Currency field Configuration Interface.

Data Type

Define the field type to Currency. It allows defining the currency number format.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	Placeholder to any character. Replaced by any character. If number of characters entered are less then the mask size, the field value is completed with zeros (Filling full size field entry is required).
Z	Replaced by any character retrieved from database. Suppress zeros at field left (Complete field filling is optional). When used combined with the mask character X it should be placed at the mask left.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

![Displaying repeated values.][cons_configuracao_geral_texto_nao_repetir_valor]

SQL Type

It informs the data type of field in the database.

Values Format

Interface of Values Format.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Currency Format

It defines if the field displays the Currency Symbol of the Regional Settings.

Currency Symbol

It allows setting the character that represents the Currency Symbol.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Display full value

It displays the full value. Example: 2018(Two thousand eighteen).

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it's value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the "Separated by" field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.

- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:

- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

In the example shown above, the width of the Chart is 200 pixels.

- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.

- **Display Value** : It will display the value and the chart at the same time, the value being on the left or right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

//## PDF Configuration {#id-06}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - Percent (Calculated) Field

General Settings

Percentage field Configuration Interface.

Data Type

Define the field type to Percentage.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Accumulated

It defines the field to accumulate values of another field. This option is only available for virtual fields.

Accumulator field

It is available when the option “Accumulated” is enabled. This configuration defines which field will accumulate the values. The field should be Integer, Currency, or Decimal.

Grid Mask

It defines a mask for the field value. There is two types of masks:

Character	Description
X	It sets the placeholder to any character for replacing. If the entered number of characters is less than the mask size, It completes the field value with zeros (Filling full-size field entry is required).
Z	It sets the placeholder to any character for replacing, but it suppresses zeros at field left (Complete field filling is optional). Use it on the left when combined with the X mask.

Mask Examples

Mask	Field Value	Formatted Value
(xx) xxxx - xxxx	1234567890	(12) 3456 - 7890
(xx) xxxx - xxxx	12345678	(00) 1234 - 5678
(zz) xxxx - xxxx	1234567890	(12) 3456 - 7890
(zz) xxxx - xxxx	12345678	( ) 1234 - 5678
(zz) xxxx - xxxx	0012345678	( ) 1234 - 5678

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the attributes Grouping, Negative Symbol and negative number format.

Digit Grouping Separator

It defines the character to separate the thousands.

Decimal Separator

It defines the decimal separator.

Negative sign

It defines the character to represent negative numbers.

Negative number format

It defines the position of the Negative sign.

Color of Negative

It defines a color for the field when the value is negative.

Decimal Precision

It defines the number of decimal places for the field.

Complete with zeros

Defines the decimal places that will or not be completed zeros to the right. Example: The value is 125,50, but the decimal precision is 3. If the option is enabled, then the value will 125,500.

Lookup Settings

A method used to offer the user a list of values that can be selected in the form applications. For example, the gender field in the Database with the values M or W can be presented like Man or Women or in a dynamic way, having these values recovered from the database.

- **Lookup Methods**

You will have available two types of lookup methods: Manual (informing manually the conditions of the select) or Automatic (using a database query).

- **Lookup Method - Automatic**

- **SQL Select Statement** : Define the SQL command that will recover the values that will be displayed on the form field. The command needs to have the following structure:

  SELECT display_field FROM table WHERE key_field = {grid_field}

  The Grid field needs to be referenced buy the curly brackets {}. While running the application, the field within the curly brackets is replaced by it’s value in the table.

- **Multiple Values** : When enabled, allows to select more than one item from the list, separated by a delimiter.
- **Delimiter** : Defines the character used to separate the values selected in the form. Only inform a value to this field if you need to separate the selected options with a different delimiter then (;).
- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value

returned from the lookup. It is only necessary to inform this value of the option Display original and lookup value is enabled.
- **Choose connection** : Allows you to select another specific connection existing in the project. The select command will be done on the second connection.

Lookup Method - Manual

This method is used so that the developer can create manually the list of values that will be saved to the generated application. This method is mostly used when the database does not have a table that contains this information.

- **Lookup Type** : Define the functionality of the lookup, being: Single Value, Multiple Values (delimiter), Multiple Values (position) and Multiple Values (binary).
- **Single Value** :

It will be displayed only one value for the field. And it will replace the stored value from the table with the one defined on the item label.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Multiple Values (Delimiter)**

It can be displayed various values for the selected field. The values should be separated by a delimiter that will be informed. It is done a replacement on the parts of the field, separated by the delimiter, with the values contained in the list.

- **Label** : Text that will be displayed in the item list of the select field.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.
- **Delimiter** Characters used to separate Multiple Values in the table field. If not informed, the default delimiter is a semi coma (;).
- **Multiple Values (position)**

Allows to store a diversity of information selected on the field in only one string. For this information to be added, you need to inform the label, initial position and the number of bytes that each information occupies on a string.

For this example, Gender, Marital status and Hobby were used. Gender uses one byte as does the Martial Status, the Hobby will use two bytes.

The example below:

Label	Value	Start	Size
Man	M	1	1
Women	W	1	1
Divorced	D	2	1
Single	S	2	1
Study	SD	3	2
Sports	SP	3	2
Read	RD	3	2

Example: Of you choose **Man**, **Single** and **Read**, in database would be stored the following value **MSRD**.

- **Label** : Text that will be displayed in the Grid.
- **Value** : Attribute that corresponds to the value that is stored in the table field. For example, the value M will be included into the table when the user will view in the list **Man**.

- **Start** : Starting position of the string that is going to be stored. The first position is always 1.
- **Size** : Amount of bytes that is going to occupy in the string.

Multiple Values (binary)

Allows to store a diversity of information selected on the field in only one value.

In this example uses the following list (although it is not informed when inserting values to the list, each item has a value applied automatically according to the order):

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

To store the data, it is done an internal sum of the decimal number. See the examples below.

Example 1: If the options **Sports** and **Culture** were selected, the number stored in the table would be **3**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 2 (Culture) = **3**

Example 2: If the options **Sports**, **Pleasure** and **Reading** were selected, the number stored in the table would be **13**.

Attribute Value	Lookup Description
1	Sports
2	Culture
4	Pleasure
8	Reading
16	Music

1 (Sports) + 4 (Pleasure) + 8 (Reading) = **13**

- **Label** : Text that will be displayed in the Grid.
- **Insert Button** : Adds to the list the values informed on the fields Label and Value.
- **Update Button** : Modifies the attributes of the selected item.
- **Remove Button** : Remove the selected item from the list.
- **Clear Button** : Clear the Fields.
- **Save Button** : Allows the user to save all the items of the list, to use on other fields, just click on Load lookup definition.

- **Load lookup definitions** : Allows to refresh the list of values predefined when using the lookup. It will be listed the existing definitions in ScriptCase and the ones saved by the user.

- **Use lookup in** :
- **Grid** : Applies the lookup to all the Grid formats (HTML, PDF, XML , CSV and RTF).
- **Summary** : Applies the lookup only for the summary (HTML and PDF).
- **Both** : Applies to both the Grid and Summary.
- **Default value** : Defines the default value, used in the cases where the select command does not return any values.
- **Display original and lookup value** : When selected **No**, its displayed on it the value from the select command. Otherwise, it will be displayed the original value of the field and the value returned from the select command separated by the character in the “Separated by” field.
- **Separated by** : Defines which character will be used to separate the values of the field with the value returned from the lookup. It is only necessary to inform this value of the option Display original and

lookup value is enabled.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Line Chart

Using this guide, you can define ac chart for each line.

- **Line Chart Type** : There are two types of charts that can be used. Below you can see their description:
- **Bar Chart** : This option allows to display a bar chart relating to the column that you are working with. The size of the bars is done by math relating to the Grand Total of the column. The value informed in the width of the chart defines the size of the images that will be displayed.

  In the example shown above, the width of the Chart is 200 pixels.

- **Line Chart** : This option allows to display a Line Chart relating to the column that you are working with . The amount of lines are done by dividing the value of a column by the value informed to the number of icons.

  In the example above, the field Number of Icons, it was informed the value 1000, so the number of stars besides, it determined by dividing the displayed value by 1000.

- **Display Value** : It will display the value and the chart at the same time, the value being on the left or right side of the bar or lines.
- **Number of Icons** : Determines the amount of icons (figures) that will appear for each value on this field. This value will be presented by the dividing the value of the column by the Number of Icons.
- **Chart width** : Width of the chart in pixels. This field is used by the option Bar Chart. In case it is not

informed, the value used is 200 pixels.
- **Chart height** : Height of the chart in pixels. This field is used by the option Bar Chart.
- **Margin** : Charts margin. This option is only used by the bar chart.
- **Image / Positive Color** : The way that it is informed will variate depending on the options below:
  - **Bar Chart** : Inform a color for the field that will be used to form the chart image. Click the icon beside to choose the color.
  - **Line Chart** : Inform an image name for the field that will be used like an icon. Click the icon beside to choose the image “Icon”.
- **Image / Negative Color** : The information passed to this field follows the same criteria as the Image / Positive Color field. This parameter will be used in the case the value is negative.

//## PDF Configuration {#id-06

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - HTML Image Field

General Settings

HTML Image field Configuration Interface.

Data Type

Define the type of field. When setting it to HTML Image, it allows to display an image into the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Image

Define an image to be displayed. The icon “Select Image” lists all images from Scriptcase and your uploaded images. The “Upload an image” option allows you to send a copy to the Scriptcase server.

New Link

This button allows the creation of a field link with some other application..

This way, it is possible, for example, to create a link with a blank to delete a record from the grid, passing the record ID as a parameter.

> See more information about field binding by clicking here.

Border

Define the width of the Image border in Pixels.

Width

Define the image width size in Pixels.

Height

Define the image height size in Pixels.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the

background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Procedure - Credit Card Number Field

General Settings

Credit Card Number Configuration Interface.

- **Data Type** : Select the type of field for the application. When it is defined as a Credit Card Number, you can define some rules for the display format of the Credit Card.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **Do Not Repeat Value** : Do not repeat the value of the field in the case it is the same as the previous record.

- **SQL Type** : Informs the data type of the field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

//## PDF Configuration {#id-03}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.

- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - E-mail Field

General Settings

Email field Configuration Interface.

- **Data Type** : Defines the type of field for the application. When it is defined as an Email, when you click on the field you be offered a choice for your email client and send an email to that specific email.
- **Label** : Defines the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Do Not Repeat Value** : Do not repeat the value of the field in case it is the same as the previous record.

- **SQL Type** : Informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

//## PDF Configuration {#id-03}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - URL Field

General Settings

URL field Configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

//## PDF Configuration {#id-03}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - YouTube Field

General Settings

YouTube field Configuration Interface.

Data Type

Define the type of field. When setting it to YouTube, it allows displaying an Youtube video on the field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Display Mode

Define the mode to display the video in the application.

Width

Video Width in pixels.

Height

Video Height in pixels.

Link Type

The display settings of the link, if it is going to be a Text or a Button.

Link text

It displays a text to mask the video URL.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.

- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Procedure - Google Maps Field

General Settings

Google Maps field Configuration Interface.

- **Data Type** : Define the type of field for the application. When it is defined as Google Maps, it will use the Google Maps API to display the map in the Grid Applications
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Home** : Defines what type of parameters will be used in the API.

Home Configuration Interface.

- **Display Mode** : Indicates the display mode of the map. It can be opened in a Modal or in a new Window.
- **Width** : Defines the width of the map that is going to be displayed.
- **Height** : Defines the height of the map that is going to be displayed.
- **Zoom** : Defines the initial Zoom (available from the Google API) of the Map location.
- **API Key** : API Key for authorization to use Google Maps in the Application. (Required only for the versions 2 or earlier of the Google API.)

  The API Key is a unique key, composed by a string(text) alphanumeric, which is the license to use the service. When you subscribe to use the service, the key is tied to the domain and the directory of the server. All the pages that use the API needs to be in the same directory that was used for the subscription. In case you have a web server on your local machine, you just need to possess a key for testing, and to do this you only need to set http://localhost in the domain of the subscription.

  To get your API Key access the site by clicking here

- **Link Type** : Defines how the link will be displayed.
- **Text Link** : Text to call the Map.

- **Marker Description** : Displays the description for each marker displayed on the map.

- **SQL Type** : Informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top,

middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Procedure - Date Field

General Settings

Date field Configuration Interface.

Data Type

Define the type of field. When setting it to Date, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

 Format of Values without Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying dates.

Detail Mask

Define the format of the field in the Grid. It uses the patterns of PHP function Date.

d-m-Y	25-09-2009
F/Y	September/2009
j/n/Y \a\s g:i:s A	25/9/2001 as 14:30:11 PM
l, d \d\e F \d\e Y	Thursday, 25 of January of 2009
h:i:s	11:33:20
#h:i:s	123:43:27 (accumulating the hours)

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

- **Example 1** : If the field is storing the Date value in a **char(8)** where the positions **1** to **4** represents the year, the positions **5** and **6** is the month and the position **7** and **8** is the day, you must define the internal format as: **YYYYMMDD**
- **Example 2** : If the field is storing the Date value in a **char(6)** where the positions **1** and **2** represents the month and the positions **3** to **6** is the year, you must define the internal format as: **MMYYYY**

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

//## PDF Configuration {#id-04}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - Time Field

General Settings

Time field Configuration Interface.

Data Type

Define the type of field. When setting it to Time, you can inform a time format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

 Format of Values without Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the Time. When not enabled, it displays the Time separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying Time.

Detail Mask

Define the format of the field in the Grid. It uses the patterns of PHP function Date.

h:i:s	11:33:20
#h:i:s	123:43:27 (accumulating the hours)

Internal Format

It allows defining the field format when the SQL type is different from **TIME**. You must use the characters **HH**, **II**, and **SS** that correspond to **Day**, **Hour**, **Minutes** and **Seconds**.

- **Example 1**: If the field is storing the Date value in a **char(8)** where the positions **1** to **4** represents the year, the positions **5** and **6** is the month and the position **7** and **8** is the day, you must define the internal format as **YYYYMMDD**
- **Example 2**: If the field is storing the Date value in a **char(6)** where the positions **1** and **2** represents the month and the positions **3** to **6** is the year, you must define the internal format as **MMYYYY**

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of

scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

//## PDF Configuration {#id-04}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - Date and Time Field

General Settings

Date and Time field Configuration Interface.

Data Type

Define the type of field. When setting it to Date and Time, you can inform its format into this field.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Do Not Repeat Value

Do not repeat the value of the field if it is the same as the previous record.

SQL Type

It informs the data type of field in the database.

Values Format

Format of Values with Regional Settings.

Format of Values without Regional Settings.

Regional Settings

It allows applying the Regional Settings to format the dates of the field. When not enabled, it displays the date separator attribute (Check the Regional Settings).

Display

It offers predefined formats for displaying dates.

Detail Mask

Define the format of the field in the Grid. It uses the patterns of PHP function Date.

d-m-Y	25-09-2009
F/Y	September/2009
j/n/Y \a\s g:i:s A	25/9/2001 as 14:30:11 PM
l, d \d\e F \d\e Y	Thursday, 25 of January of 2009
h:i:s	11:33:20
#h:i:s	123:43:27 (accumulating the hours)

![Date Format Table][cons_data_format_date]

Internal Format

It allows defining the field format when the SQL type is different from **Date**. You must use the characters **Y**, **M** and **D** that correspond to **Year**, **Month** and **Day**.

- **Example 1** : If the field is storing the Date value in a **char(8)** where the positions **1** to **4** represents the year, the positions **5** and **6** is the month and the position **7** and **8** is the day, you must define the internal format as: **YYYYMMDD**
- **Example 2** : If the field is storing the Date value in a **char(6)** where the positions **1** and **2** represents the month and the positions **3** to **6** is the year, you must define the internal format as: **MMYYYY**

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

//## PDF Configuration {#id-04}

Chart Settings

By using this option, you will see an icon on the Field Title, this icon displays a chart relating to the field.

On this type option you need to choose two columns to build the chart. The first one is the field its self that you are working with, and the second one is the you chose and is called “Column for Label”.

Bar Chart Interface.

- **Group by Label** : Groups the values of the column by the label. Similar to the effect of the group by of the a Select.
- **Summary Function** : Function that summarizes the data applied to the column. The functions that can be used on the field are the following: Count, Sum, Max, Min and Avg.
- **Configurable Chart** : Allows the user to setup the parameters of the chart creation when the application is running.
- **Display Values** : Displays the values of the generated chart.
- **Column for Label** : Selecting the column that will be the label for the field.
- **Chart Width** : Width of the chart, in pixels.
- **Chart Height** : Height of the chart, in pixels.

Chart Layout

Chart layout settings interface.

Procedure - Image (Database) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Image (Database), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Image Border

Width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Open in Another Window

Allows to open the image in another window.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.

- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Procedure - Image (File Name) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Image (File Name), all the Images files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Image Border

Width of the Image border in Pixels.

Image Height

Image height size in Pixels.

Image Width

Image width size in Pixels.

Maintain Aspect

It maintains the original aspect ratio of the image when resizing it.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API.

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API.

Open in Another Window

Allows to open the image in another window.

Subdirectory for local storage

Local subdirectory or on the server where the files are stored. eg: {CustomerId} , [glo_var_seq]

Image Caching

Time in minutes the image cache will be kept in the server before being deleted.

Repeat value

Repeat the field value if it is equal to the previous record.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Procedure - Document (Database) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Document (Database), all the document files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Icon

Displays an icon beside the field to identify the type of document.

File Name

It allows defining the field to store the name of the document in the database.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.
  - **Title background color** : Allows choosing a color for the background of the field title using a color palette.
  - **Bold** : Displays the field title in bold

Procedure - Document (File Name) Field

General Settings

Upload field Configuration Interface.

Data Type

Define the type of field. When setting it to Document (File Name), all the document files are stored and loaded directly from the Database.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

Sub-folder

Sub-folder name that the files are stored.

Icon

Displays an icon beside the field to identify the type of document.

File Name

It allows defining the field to store the name of the document in the database.

Cloud storage API

Profile with the API information (Dropbox, Google Drive or Amazon S3) used to upload the file to the cloud created in Tools > API.

Storage API directory

Directory name for upload to the Cloud Storage API. Eg.: files/docs/{id}. The file will be saved in the root folder if this field is empty.

Storage API cache directory

Local directory for caching files sent to the cloud via the storage API.

SQL Type

It informs the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or

bottom).
- **Background Image** : Allows you to add a background image to the field.
- **Line Break** : Enables the possibility of a line break for the field.
- **Content color** : Changes the content color of the query field and the grouped field, when available.
- **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Procedure - Bar code Field

General Settings

Configuration Interface of the Barcode Field.

- **Data Type** : DataType of the field for the application.
- **Label** : Set the title of a field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name would be “Client Name”.
- **SQL Type** : Database field type.

Watch below a video showing an example about the Barcode field:

Values Format

Configuration Interface of the Barcode Field.

- **Type** : Type of Barcode.
- **Text** : Barcode Text for illustration purposes.
- There are **18 types of barcodes**, that are listed below:

Barcode configuration interface.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.
  - **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
  - **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
  - **Background Image** : Allows you to add a background image to the field.
  - **Line Break** : Enables the possibility of a line break for the field.
  - **Content color** : Changes the content color of the query field and the grouped field, when available.
  - **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
  - **Width** : To define the width of the field.
  - **Height** : To set the height of the field.
  - **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
  - **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
  - **Title color** : Allows you to choose a color for the field title using a color palette.

- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Procedure - QRCode Field

General Settings

QRCODE field Configuration Interface.

Data Type

Define the type of field. When setting it to QRCODE, it allows you to transform values into a QRCODE.

Label

The Label option lets you define the title of a field. Example: If the database field name is “cmp_name_client”, You can display a different name for the user, like “Client Name”.

SQL Type

It informs the data type of field in the database.

Watch below a video showing an example with QRCODE

Values Format

Interface of Values Format.

Level of error correction

The Codewords are 8 bits long and use the Reed–Solomon error correction algorithm with four error correction levels. The higher the error correction level, the less storage capacity.

Image Size

Set the size of the QRCODE image.

Margin

Set the margins of the QRCODE.

 Interface of Values Format.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of scriptcase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **Display Settings**
  - **Text Font** : Allows choosing the font type, which will be applied in the application field.
  - **Font Size** : Allows choosing the font size, which will be applied to the application object.
  - **Italic Text** : Allows to apply the italic type in the font.
  - **Bold Text** : Allows you to apply bold type to the font.
  - **Number of characters** : Number of characters to display.

- **Text Alignment** : Allows you to position the field in the desired location (left, right, center and justified).
- **Vertical Alignment** : Allows you to position the field in the desired location (top, middle or bottom).
- **Background Image** : Allows you to add a background image to the field.
- **Line Break** : Enables the possibility of a line break for the field.
- **Content color** : Changes the content color of the query field and the grouped field, when available.
- **Background color** : Allows choosing the color, using a color palette to be applied as the background of the application field.
- **Width** : To define the width of the field.
- **Height** : To set the height of the field.
- **Title Horizontal Alignment** : Allows you to define the horizontal alignment for the field title (left, right or centered).
- **Title Vertical Alignment** : Allows you to define the vertical alignment for the field title (top, middle or bottom).
- **Title color** : Allows you to choose a color for the field title using a color palette.
- **Title background color** : Allows choosing a color for the background of the field title using a color palette.
- **Bold** : Displays the field title in bold

Procedure - Signature Field

General Settings

Configuration Interface of the Signature Field.

The signature field will help you creating more sofisticated forms and making it possible to store signatures in your database. Inside our development environment we have specific settings that will help you to customize your field, those options are:

- **Data Type** : You can define the type of field for the application. When it is defined as a text, it accepts letters, numbers and special characters.
- **Label** : Lets you define a label to the field in the application. For example: if the field is defined in the database as cmp_name_client, it will be easier to the user understand if the name is “Client Name”.
- **Label below field** : Defines the message to be displayed below the field.
- **Background color** : Defines a color to the field background by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Pen color**: Set a color to the pen by using a valid hexadecimal color value or by choosing a color from the color palette.
- **Width** : Lets you define a width to the field.
- **Height** : Set a height to the field.
- **Subtitle** : Defines the subtitle that will be displayed beside the field.
- **Initial Value** : Lets you define the initial value for the field when in insert mode. The options are:
  - **Defined Value** : The field will receive the value from the text field of the defined value option.
- **Save Variable** : Save a variable in the session with the value of the field, that can be used in other applications. For example, in the login form the user name can be saved in the session so that it can be displayed on the header of other applications.
- **Variable Name** : Set the name for the session variable that will receive the field value.
- **Disabled Field** : Define if the field will be disabled in “Insert Mode”, “Update Mode” or in “Insert and Update Mode”.
- **HTML Type** : Displays the HTML Object that will be used to display the field in the form.
- **SQL Type** : Displays the data type of field in the database.

Display Settings

The Display Settings allows to define the CSS values individually for each field. For each Display schema of ScriptCase, there are the same attributes available in Interface.

Display Settings configuration Interface.

- **CSS of the Title**
  - **Font** : Select the font type, that will be applied to the application field title.
  - **Font Size** : Defines the font size, that will be applied to the application field title.
  - **Font Color** : Choose a color to the font by using a valid hexadecimal color value or from the color picker.
  - **Background Color** : You can define the color for the field by using a valid hexadecimal color value or from the color picker.
  - **Bold** : Applies the bold style to the font.
  - **Underline** : Set the underline style to the font.
  - **Border style** : Choose a style for the border.
  - **Collapse** : Defines the collapse for the border.
  - **Border Size** : Applies the border size to the title of the field.
  - **Border Color** : Choose a color for the border, using a color palette to apply to the title.
  - **Horizontal Alignment** : Position the label of the field in the wanted position (left,right,center and justify).
  - **Vertical Alignment** : Position the label of the field in the wanted position (baseline, sub, super, top, text-top, middle, bottom, text-bottom).
  - **Width** : Defines a width for the title of the field.
  - **Height** : Set a height for the title of the field.

Procedure - Rating Field

This field allows the developer to create a field to display the data previously added using stars (or any other image).

Attribute descriptions

Data Type

Define the field type to **Rating**.

Label

The Label option lets you define the title of a field. Example: If the database field name is “**Stars**”, You can display a different name for the user, like “**Stars**”.

Besides use a fixed text, the **Label** attribute allows the use of **Langs** to define the field title, allowing the internationalization of your application.

Subtitle

Define the subtitle of the field, below the ratings. **Example**: “Thank you for your feedback!”.

As in the **Label**, the **subtitle** attribute also allows the use of **Langs** for the internationalization of your application.

Amount of icons

It defines how many icons it will display in the field. The value set in this attribute must be according to the evaluation rules.

Use in Group by and Sums

Allows the field to be used instead of the value when the field is selected in the **(static or dynamic)** group by or in the totalization of the summary. To do this, just add the fields in the dynamic or static group by column to be able to visualize the grouping performed in the query.

Example of the **Rating** field in the summary.

In the summary, when hovering the mouse over the field, a small window detailing the ratings that were carried out both in each record and in total will be displayed, as shown in the image below.

Display absolute value

Allows the exact number of ratings that each star received to be displayed in the detailing window, next to the percentage, when summing up the summary.

> With this flag turned off, the grouping will only display the voting percentage, leaving out the exact number of ratings.

In the case of the result of the ratings, this result is calculated through the weighted average **(Sum/Total of Evaluations)** as in the example below:

For example: in the image above, we have a total of 4 ratings carried out, where:

0 rating for 5 stars
2 ratings for 4 stars

0 rating for 3 stars
1 rating for 2 stars
1 rating for 1 star

Considering that the number of stars refers to the weight each one has, we have the following:

5 stars = 0 rating x 5 (weight) = 0
4 stars = 2 ratings x 4 (weight) = 8
3 stars = 0 rating x 3 (weight) = 0
2 stars = 1 rating x 2 (weight) = 2
1 star = 1 rating x 1 (weight) = 1

Sum = 11

We took the result of the sum and divided it by the total number of evaluations, which was 4.

Sum/Total of Evaluations = 11/4 = 2.75 ≈ 2.8.

As only 1 value after the comma will always be displayed in the detail window, rounding will occur upwards when the values after the comma are above the central value, and downwards when it is less than the central value.

Ex.: 2.75 ≈ 2.8.

Use Font Awesome

Allows you to use Font Awesome icons to represent stars instead of images.

Rating Star Hints

Defines the hint for each star

Icon size

Sets the pixel size of star icons.

Rating Padding

Defines the padding value in pixels for separating each star.

Icons Color

Allows you to define the color of the stars; the color is passed in hexadecimal.

SQL Type

It informs the data type of field in the database.

Configuração da Visualização

You can set CSS values for the field individually. Thus, for example, you can highlight one field of the others in the application. As CSS field properties, when changed, they are added to a class created automatically by ScriptCase for each application field.

> Individual field CSS settings, when inserted, override theme settings (CSS of applications (Themes)) selected for application.

The settings are divided into three property blocks, these blocks are:

CSS of Title

This makes it possible to change the CSS properties of the field's Label.

In the example below, You can see the difference of field title configurations. While the field **Contactname** have the same formatting, inherits the theme of the application, the field **Contacttitle** have a different formation of of others, from the changes made in the CSS of the field.

CSS of field

Changes the CSS properties of the <td> where the input object (where the user enters data for insertion into a form) is positioned. In the image below, you can see where the change is applied.

Field Contacttitle with changes to Field CSS properties, changing background color and horizontal alignment

CSS of Input object

Changes the CSS properties in the Input of field object, where the user type the data in a form.

Field with changes to CSS properties of Input Object, changing background color and input source color

CSS properties

The available configuration options are basically the same for each of the configuration blocks above.

description of available configuration attributes

Font

Changes the font of the text according to the fonts selected using the *font-family* property.

In this option, some types of fonts are provided to you (as shown below).

Font Size

Changes the font size of the text using the property *font-size* in the field class.

You need to select the available value from our list, the measure used for this property is the pixel.

Font Color

Define the font color used by the property *color* in the field class.

Background color

Define the background color using the property *background-color* in the field class.

The colors that will be used in the two color properties listed above, **Font color** and **Background color**, can be entered via the color palette - - available next to the field or manually entered values in the accepted formats that are: *Hexadecimal*, *RGB*, *RGBA*, *HSL*, *HSLA* or o *Color name*.

Color palette
By clicking on the color palette icon - - next to the field, a window will open with some default colors.

When you select one of the colors, a value in hexadecimal format (HEX) will be entered, representing the chosen color.

Hexadecimal
Acronym for hexadecimal, this code is composed of the pound sign (#) plus six digits. The first two define the intensity of the color red, the middle two are green and the last two are blue.

Bold

Applies bold style to the font.

Underline

Lets you apply the underlined style to the font.

Border style

Defines the border font style.

Border Collapse

Defines the border collapse.

Border size

Changes the size of the title border.

Border color

Chooses the border color , using a color palette to apply to the title.

Horizontal alignment

Position the filter label at desired location (left,rigth,center e justify).

Vertical alignment

Position the label of the filter in the wanted location baseline, sub, super, top, text-top, middle, bottom, text-bottom).

> Not available in Input Object CSS

Width

To set the width of the title.

Height

To set the height of the title.

Allow the developer to set up instructions and hints about the field’s use, allowing the developer to help the users about the system usage.

Description of the settings attributes.

Help Description

In this area, you can set up a text that will be shown in the application according to the type of help selected below.

Help Type

Hint

Display a default icon **?** beside the field. When the mouse is above the icon a help text will be displayed.

Text

Display a text inside of a beside the field.

Pop-up

Display a default icon **?** beside the field. To display the message just click in the icon that a Pop-up will be displayed with the help text.

> The icon displayed when selecting the types **Hint** or **Pop-up** vary according to the theme that is being used in the application. This icon can be modified in the CSS Buttons.

Tippy

Displays a standard **?** icon next to the field. When you hover the mouse over the icon, the message will be displayed in a more user-friendly way.

Help type settings (Tippy)

- **Help Description** : Defines where the field’s help button will be displayed
- **Help button position** : Defines the display position of the Tippy hint in relation to the field’s help icon
- **Time to open hin Tippy** : Defines the time in milliseconds for the Tippy hint to be displayed after the mouse passes over the help icon.
- **Time to close hin Tippy** : Sets the time in milliseconds for the Tippy hint to be hidden after the mouse leaves the help icon.
- **Tippy hint width** : Sets the width of the Tippy hint in pixels.

Help button position (except Tippy)

Allow to set up where the help button will be displayed in the field, there are two options:

Beside: The help icon will be displayed beside the field.
Column: The help icon will be displayed beside the label of the field.

Procedure Layout Setting

This module lets you editing display options for the application. You can apply visualization themes, organize blocks, define values, and the display format of the Header and Footer.

Blocks

Blocks are “containers” where you can position the application fieldSlides of Forms, Controls, or Grids.

Scriptcase creates applications with one block by default. You can add more blocks as you wish, to organize it in the best way.

See below, the Columns Organization, and where you can define the position of the next block: beside or below the current one.

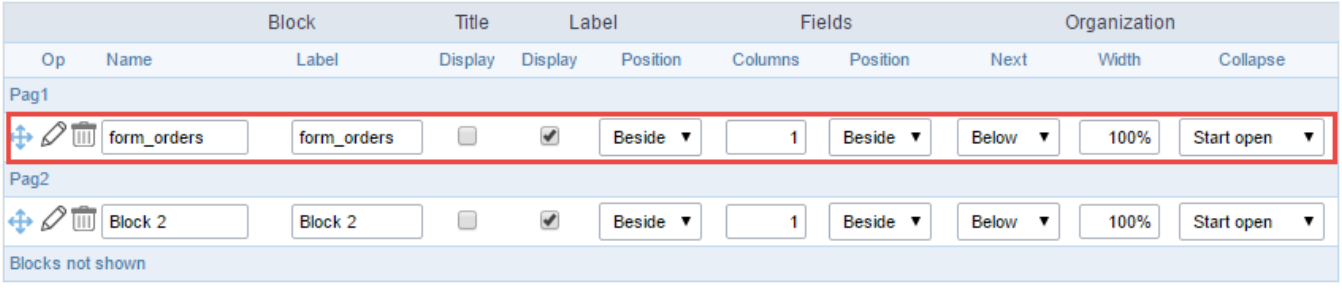
Application Block configuration

On the left side of each block, there are two icons, the first one to edit the information of the block and the second one to delete the block.

Organizing the position of the Blocks

See below how to modify the display order of the Blocks in one Page.

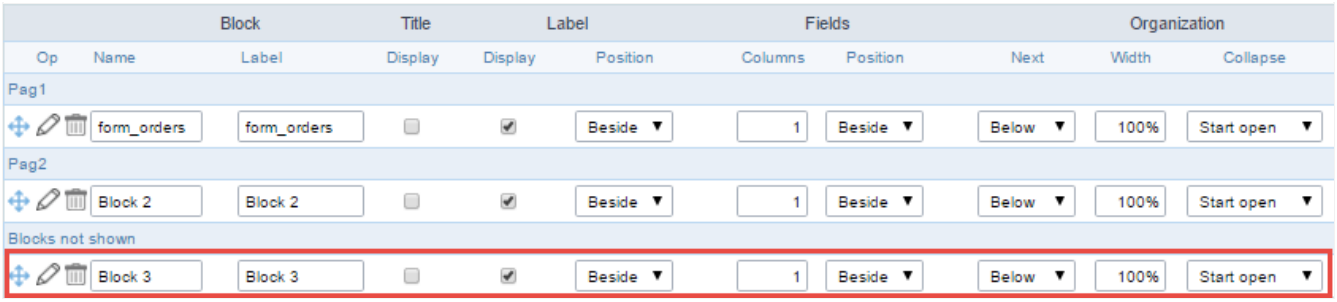
Click and drag the block that you desire to modify to its new position.



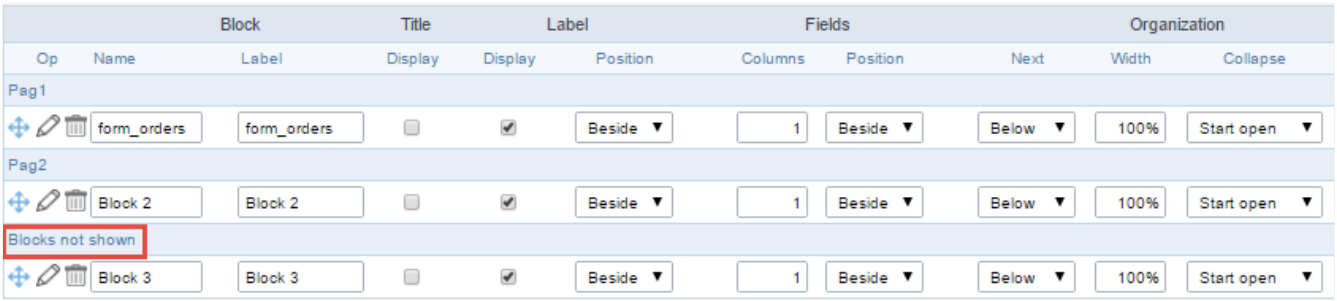
Application Block Display configuration

- See how to remove a block from the display

Click on the block desired and drag it to the item “Blocks not Shown”. This way, you can also drag the block to another page if desired. See the images below.



Application Block Display configuration



Application Block Display configuration

Block

- **Name**: The name that identifier the Block.
- **Label**: Title of the block to display in the application.

Title

- **Display**: It controls the display of the block title.

Label

- **Display**: It controls the display of the field labels of the block.
- **Position**: Options to display label :
    1. **Above**: Display the label above the field.
    2. **Beside**: Display the label beside the field.
    3. **Below**: Display the label below the field.

Fields

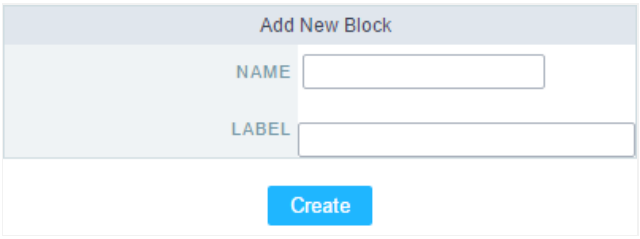
- **Columns**: Amount of columns side by side in the block.
- **Position**: The way to display the fields in the block:
    1. **Below**: Display the fields one below the other respecting the number of columns.
    2. **Beside**: Display the fields one beside the other respecting the number of columns.
    3. **Line**: Display the fields one beside the other with no tabulation.

Organization

- **Next**: The way to display the blocks in the page:
    1. **Below**: Set to show the following block below the current one.
    2. **Beside**: Set to show the following block beside the current one.
    3. **Tabs**: Set to show the following block in a different tab then the current one.
- **Width**: Set the block width in pixels or percentages. Use the symbol “%” to indicates the value in percentage.
- **Collapse**: Enables the option to close the block.

Create a New Block

To include new blocks in an Application, click on the button Create New Block. Then, enter the name and label of the block in the following interface and finish by click on Create.



Creating application blocks configuration

Name

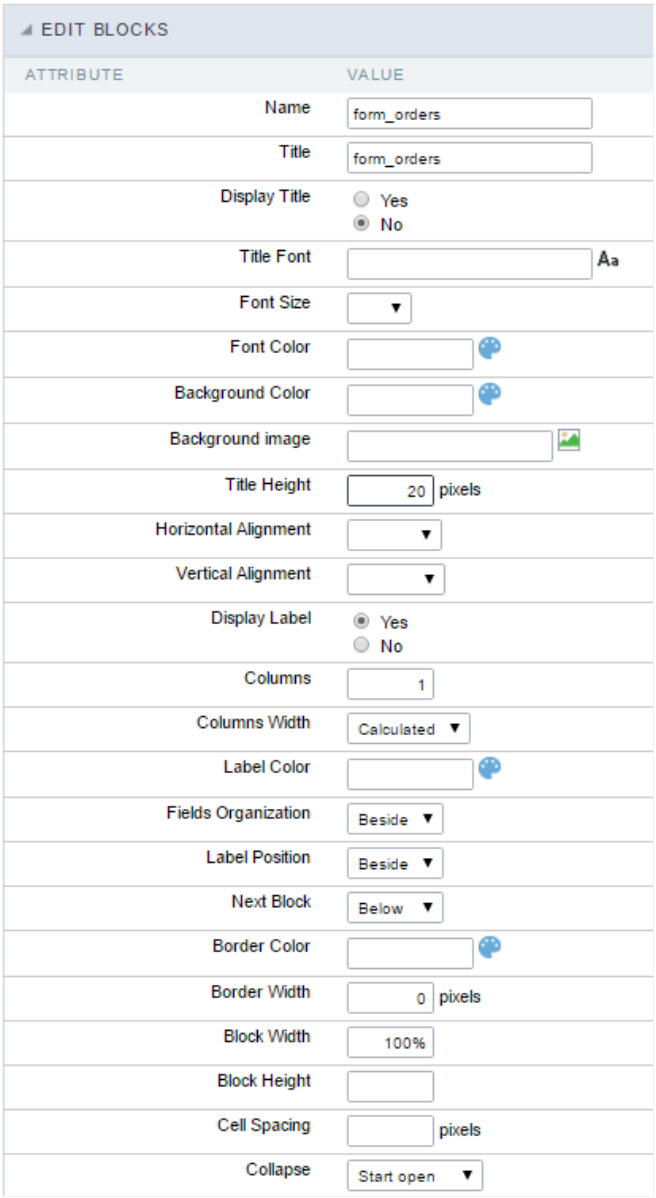
Name of the Block.

Label

Title of the block to display in the application.

Edit Blocks

To edit a block, click on the icon , that is on the left side of the block. Then you can see the following interface to define the parameters of the blocks. Click on Save to finish.



Application Block editing interface

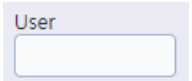
Name

Name of the block. #### Title
Block title for display. #### Display Title
This option, when active, allows displaying the block title. #### Title Font
Set the font family of the block title. #### Font Size
Set the font size of the block title. #### Font Color
Set the font color of the block title. #### Background Color
Set the Background Color of the block title. #### Background image
Set a Background image for the block title. #### Title Height
Height in pixels of the block title line. #### Horizontal Alignment
Horizontal Alignment of the block title (Left, Center, and Right). #### Vertical Alignment
Vertical Alignment of the block title (Top, Middle, and Bottom). #### Display Label
Display the labels of the fields in the block. #### Columns
Amount of field columns in a block. #### Columns Width
Set the field column width of the block. #### Label Color
Color of the field labels. #### Fields Organization
The way to display the fields in the block. #### Label Position
Set the position of the field labels of the block.

The options are:

- **Beside** - This option positions the label on the right side of the field.

- **Above** - This option places the label above the field.



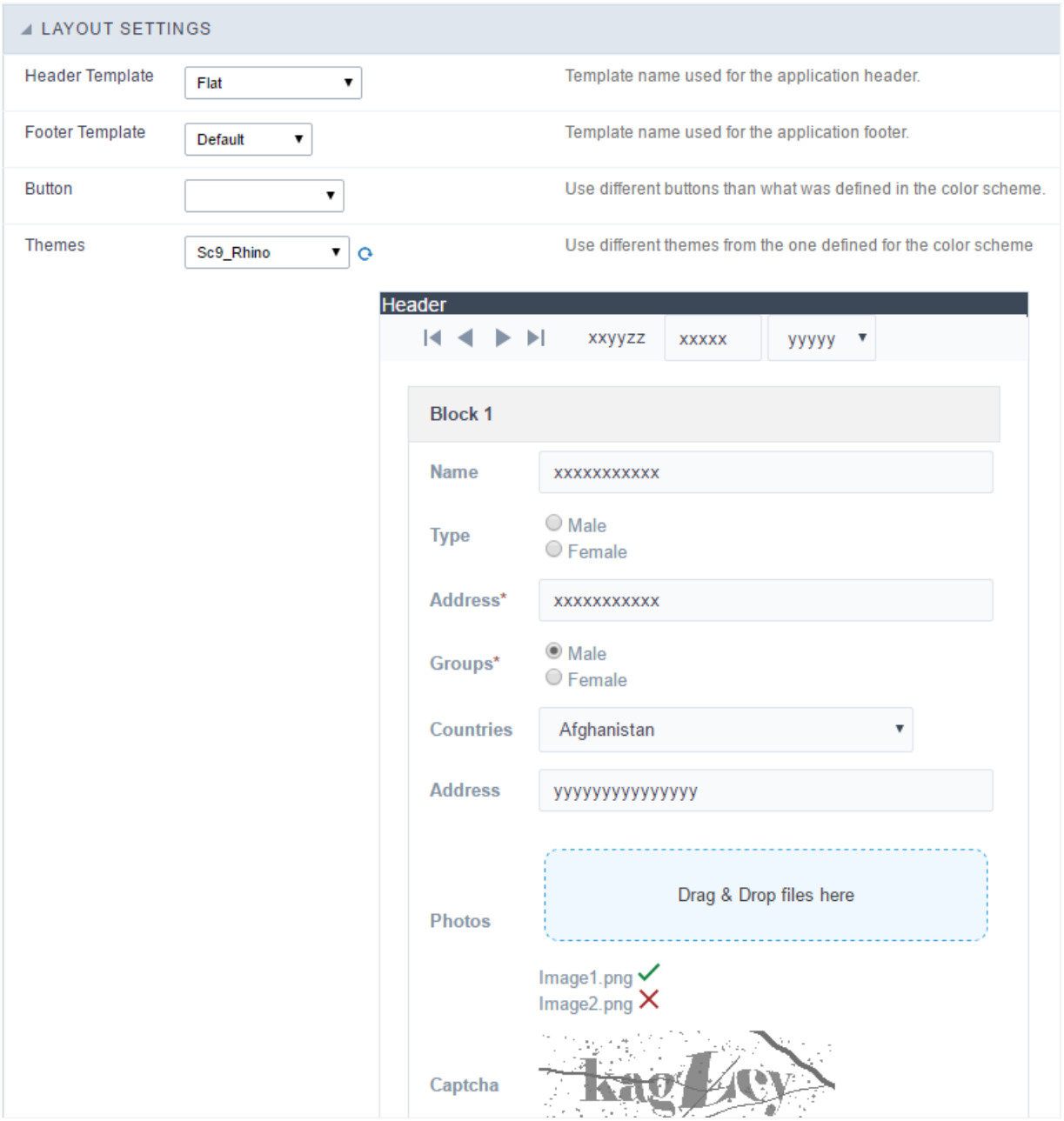
- **Below** - This option places the label below the field.

Next Block

Set the position of the following block relating to the current one. #### Border Color
The border Color for the block. #### Border Width
The border Width for the block. #### Block Width
The width for the block. #### Block Height
The Height for the block. #### Cell Spacing
The Cell Spacing in the block. #### Collapse
It enables the option to close the block.

Layout Settings

ScriptCase creates an app by using default values defined per project, but you can change those values for a specific application on this interface.



Application Layout

Header Template

Allows choosing the look of the application Header.

Footer Template

Allows choosing the look of the application Footer.

Button

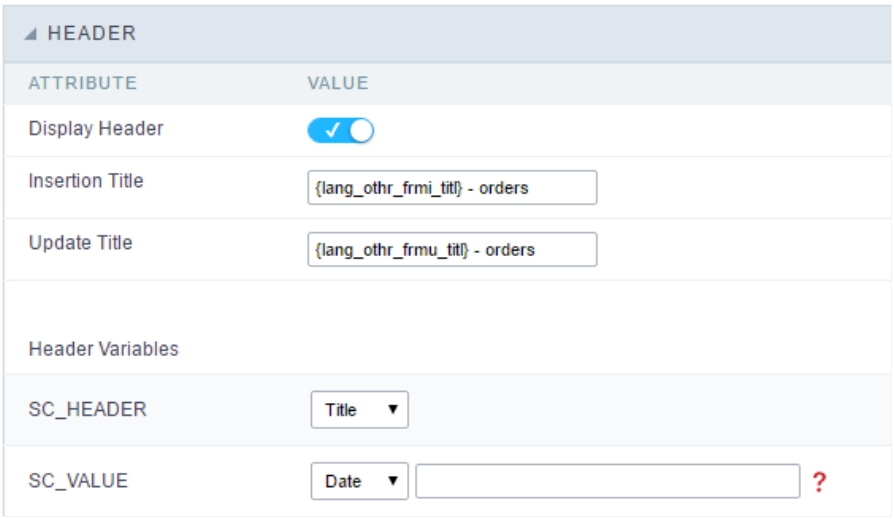
Allows choosing the button theme for the application.

Themes

Choose one of the selected themes in the project properties. It defines the look of the application, like colors, fonts, and others.

Header

Here we can define if we want to display the header or not, the application title, and fill the header variables. The header looks according to the selected template in the Layout settings interface.



Application Header Configuration

Display Header

This option determines if we want to display the header or not.

Title

Allows informing the title of the application.

Header Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

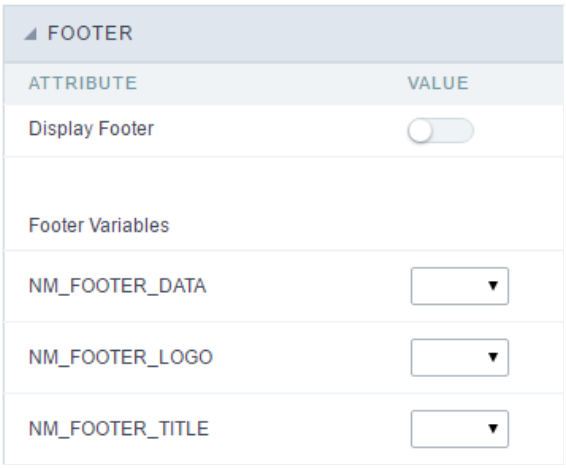
- **Field**: When you choose the option "**Field**", it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Title**: It displays the value of "**Application Title**" in the header.
- **Date**: It displays the system's date in "yyyy/mm/dd" format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the

icon “**Choose Image**”, and you still can upload new images by using the button “**Upload**”. .

- **Value**: It displays the content of the text input. You can inform static texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

Depending on the Application, you may have more than one title option.

Footer



Application Footer Settings

This page may change depending on the footer template chosen in the Layout Settings.

Display Footer

This option determines if we want to display the footer or not.

Footer Variables

We must fill the variable fields with one of the types available in the Combobox. Depending on the type, you need to associate content with it. Those are the types:

- **Field**: When you choose the option “**Field**”, it opens a Combobox to choose a field to associate the field value with the header. This option will be enabled in the applications that you can crate a fields
- **Date**: It displays the system’s date in “yyyy/mm/dd” format in the header. There are several display formats available, and you can change it by using the text field beside it. To access existing formats, click on the hint to see an explanation.
- **Image**: It displays a field to inform the name of an existing image in the server. You can select an image by clicking on the icon “**Choose Image**”, and you still can upload new images by using the button “**Upload**”. .
- **Value**: It displays the content of the text input. You can inform static texts and “**Global Variables**”. e.g. “Employee Name: [v_name]”.

Procedure Events Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application's CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event's PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Stored Procedure

It is the first event executed in the application, being fired even before the SQL and HTML assembly.

It works as a preparation for the application, where it is possible, among other things, to manipulate variables, perform validations and change the connection with the sc_change_connection macro, for example.

In the example below, access to the edit form (form_orders) linked to the query will be limited, depending on the user who is accessing the application.

If the variable [usr_login] is equal to **admin**, the user will have access to the form with the possibility of inserting and deleting a record.

If the variable has some other value, the user will still have access to the form, but only with the possibility of changing a record.

Samples Code

```
if ( [usr_login] == 'admin') {

  sc_apl_conf("form_orders", "insert", "on");
  sc_apl_conf("form_orders", "delete", "on");

} else {

  sc_apl_conf("form_orders", "insert", "off");
  sc_apl_conf("form_orders", "delete", "off");

}
```

onNavigate - Stored Procedure

In the query application, this event runs in two situations:

- When configured with partial paging, the event is executed when navigating between pages using the **navigation buttons**, **navigation by page** or the **Jump to** option.
- When configuring the application with **Infinite Scroll**, it is executed every time records are loaded.

Unlike other events, this one depends on user interaction to be triggered, allowing, for example, that any validation or layout change is performed only after this interaction.

> **Note**
>
> The query navigation is performed by ajax, thus, it is not possible to use Javascript codes (JQuery, Ajax).
>
> When using full paging in the application, the event is not executed.

onScriptInit - Stored Procedure

Second event to be executed, before the execution of the main select, it is triggered whenever the application is loaded. For example, after using the advanced filter.

In it, all application variables and libraries are available for use.

The manipulation of connections with sc_connection_edit and sc_connection_new macros, manipulation of the main select with sc_select_order and sc_select_where(add) macros and inclusion of libraries already incorporated into Scriptcase such as Jquery with sc_include_lib macro are some of the examples of using the event.

In the example below, access to the sales report will be limited to the user who performed them.

Access is limited if the user is not the admin.

Sample Code

```
if ( [usr_login] != 'admin' ) {

   if ( empty({sc_where_atual})){

      sc_select_where(add) = "WHERE employeeid > [usr_login]";

   } else {

      sc_select_where(add) = "AND employeeid > [usr_login]";
   }
}
```

onRecord - Stored Procedure

This event is executed immediately before the display of each record displayed in the query, regardless of user interaction.

Thus, it is commonly used for data manipulation and validation, creating links with the sc_link macro or changing the layout with the sc_field_style macro, based on the displayed information.

In the example below, the style of the text in the {priceorder} field will change according to the value.

For values below 500 reais, the text will have a different color from values above 500 reais,

Samples Code

```
if ( {priceorder} < 500 ) {

  sc_field_style({priceorder}, "Background-Color", "15", "#228B22", "", "bold");

} else {

  sc_field_style({priceorder}, "Background-Color", "15", "#006400", "", "bold");

}
```

onHeader - Stored Procedure

The third event generated in the query occurs immediately before the application's header display, during the loading of the page's display HTML elements.

> The onHeader is only obtained when the option to display the header is enabled in the application's layout settings.

In this event, it is possible to perform queries in the database, allowing you to read information about the application and the values of the Query's fields.

It is used, among other things, to change the application's CSS, javascript codes and display information linked to fields, such as a sales total or a legend for a line graph.

onFooter - Stored Procedure

This event occurs after the processing of the grid lines, during the display of the HTML elements of the page footer.

onFooter is used when we want to display some information in the footer, such as a sales total or a legend for a line graph.

> onFooter runs only when the option to show footer is enabled in the application's layout settings.

E.g: We want at the bottom of the application to display the total value of orders displayed in the grid of an invoice, with a 10% discount:

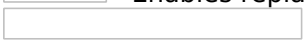
```
sc_lookup(ret," select sum( valueitem ) from sales_item where sale_id={sale_id} ");

[v_footnote] = {ret[0][0]} * 0.1;
```

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

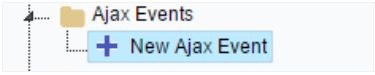
> Click Here to view the Scriptcase hotkeys documentation.

Procedure Application Ajax Events

OnClick

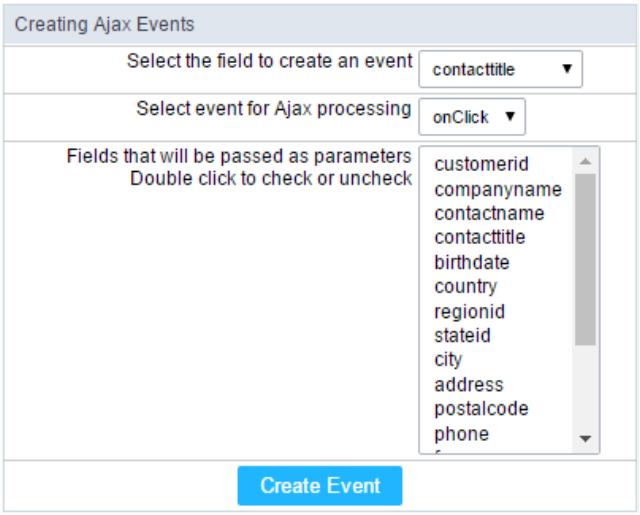
The ajax event OnClick is executed when the field that it's based on is clicked.

- Creating a new ajax event



Creating a new ajax event

- Selecting a field

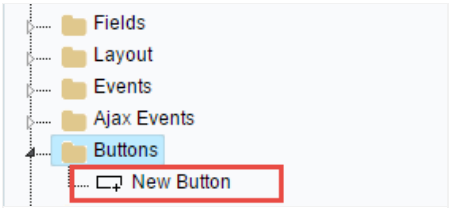


Selecting a field of the ajax event

Choose a field to create an event To define in which field the event will be add to.
Choose an event that run the ajax Defines which event will be added to the field.

Procedure Buttons Settings

In addition to the buttons that comes automatically with the applications, you can also create your own buttons. All the buttons are displayed within the application toolbar.

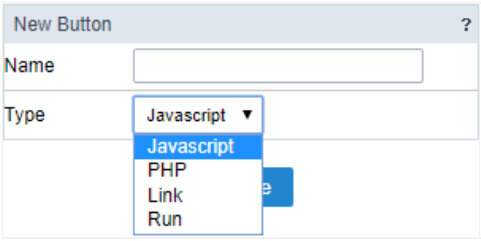


New buttons creation settings

Creating a new button

To create a new button, click on the “New Button” option and enter a name and a button type.

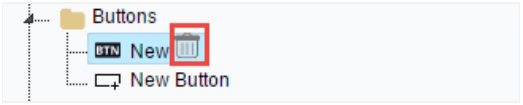
The button Types are: JavaScript, PHP, Link and Run.



Grid buttons type

Deleting a button

To delete a button click on Delete icon (represented by a recycle bin image) in the right of the button name, at the application menu under the Buttons option.



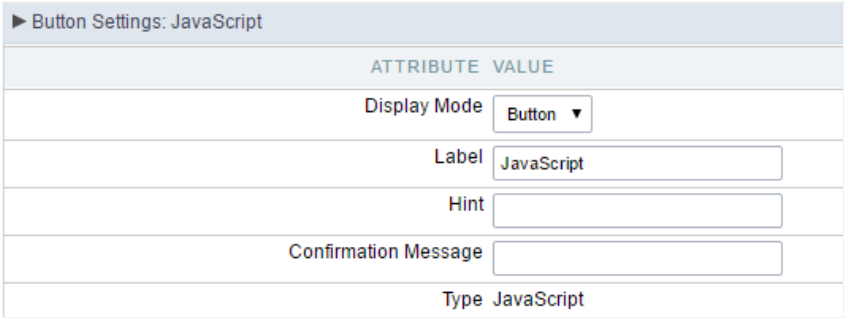
Deleting a button

JavaScript

Display Mode

You can configure the display mode of the javascript button in Image, Button or Link.

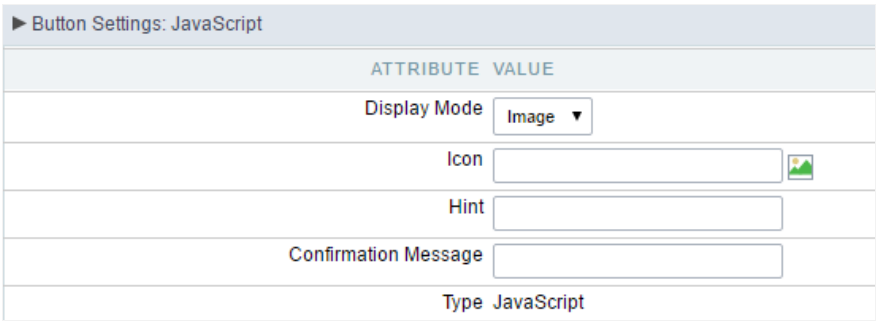
Button



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

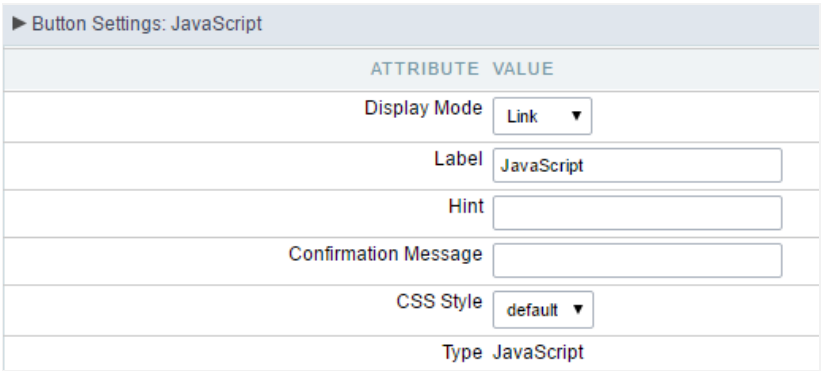
Image



Setting up Javascript Button.

Display Mode	You can select the display mode for the javascript button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

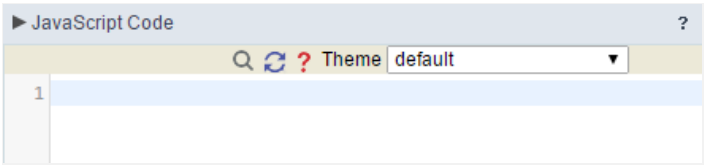
Link



Setting up Javascript Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Code Block



Javascript button coding block.

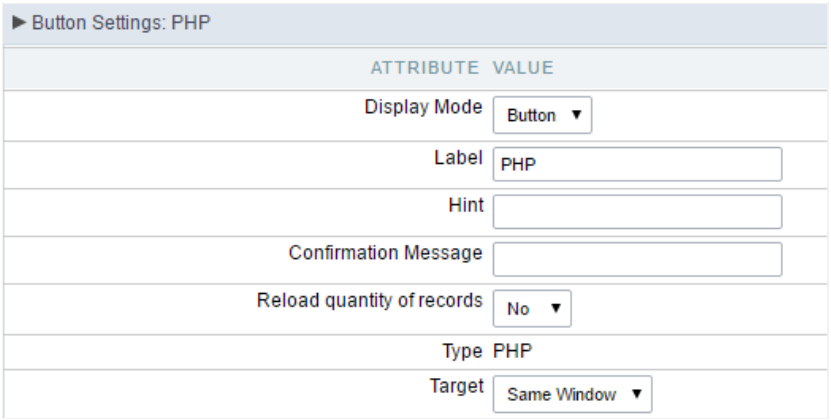
In this block, only JavaScript is accepted.

PHP

Display Mode

You can configure the display mode of the PHP button in Image, Button or Link.

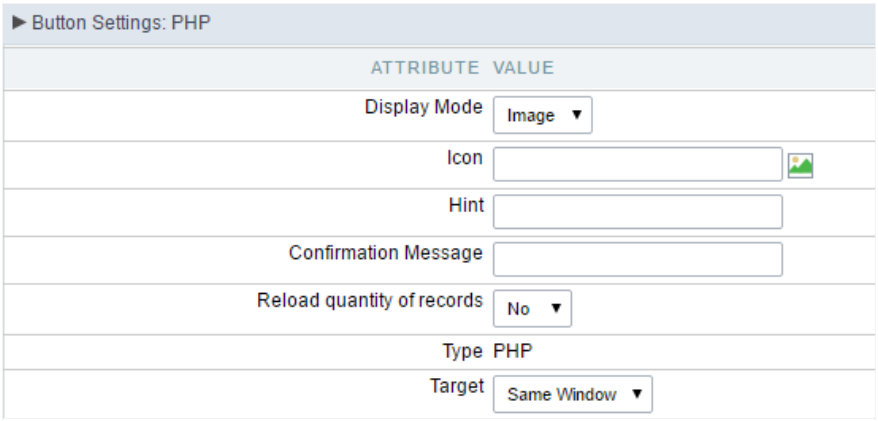
Button



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

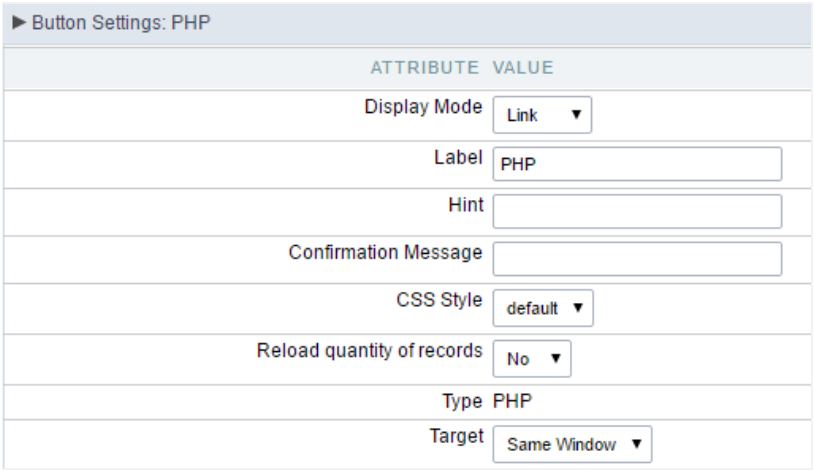
Image



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

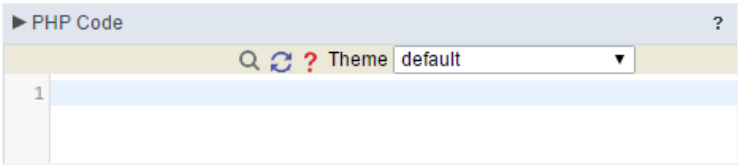
Link



Setting up PHP Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Reload quantity of records	This option is used to update the amount of records in the application.
Type	Description of the created button.
Target	Defines the window destination where the code will run (Open in the same Window, Open in another Window, Modal).

Code Block



*Ajax button coding block. *

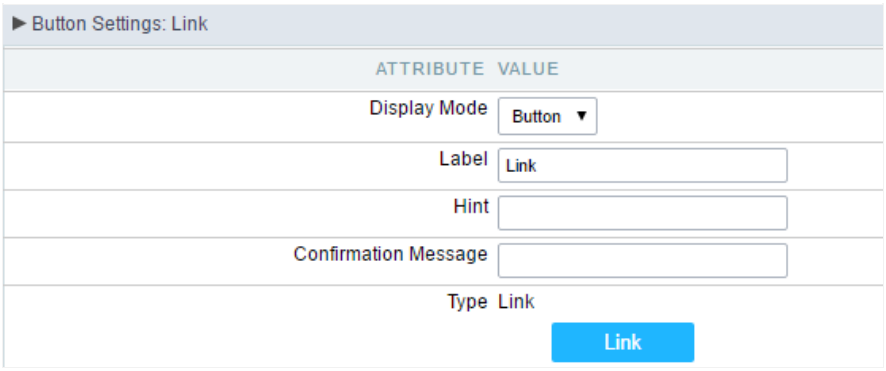
In this block, you can use macros, PHP code and JavaScript.

Link Button

Display Mode

You can configure the display mode of the link button in Image, Button or Link.

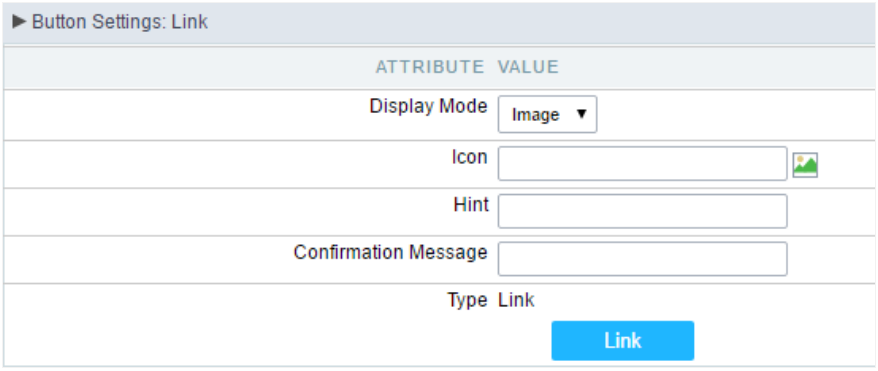
Button



Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Image

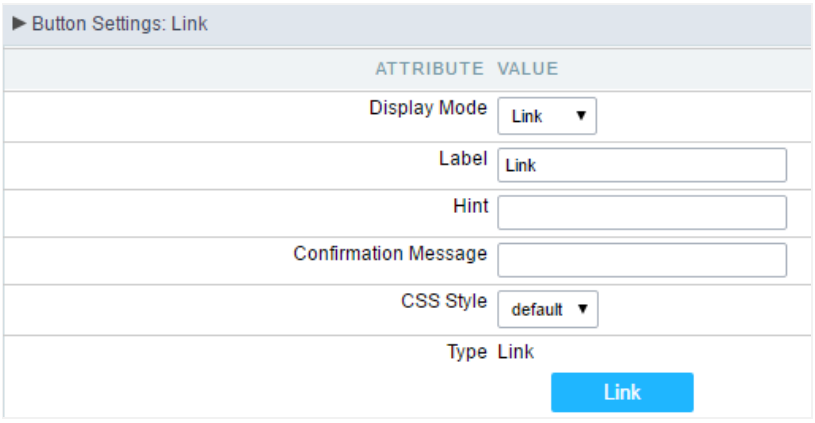


Setting up Link Button.

Display Mode	You can select the display mode for the link button in this option.
Icon	Allows you to inform the icon that will be displayed on the button while the execution of the application.

Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
Type	Description of the created button.

Link

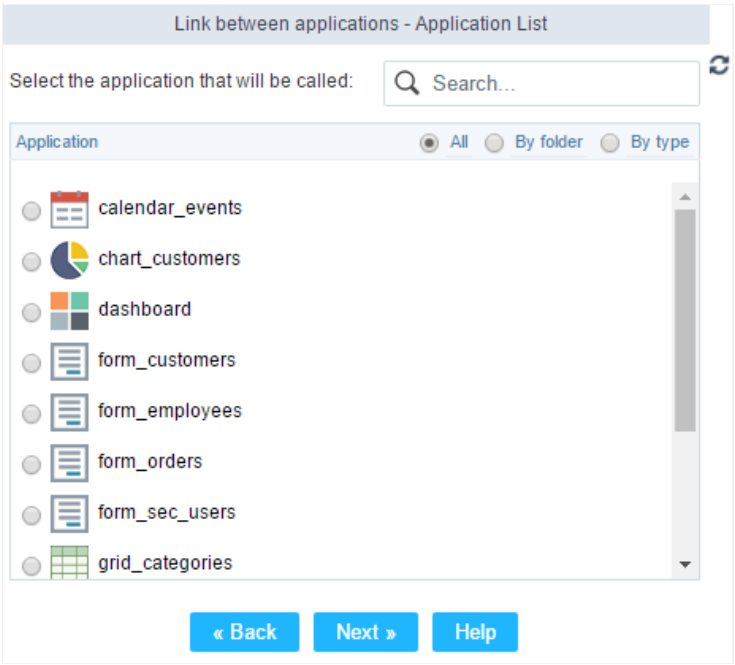


Setting up link Button.

Display Mode	You can select the display mode for the ajax button in this option.
Label	Text of the button that will be display in the application while executing.
Hint	Hint message for the button. (Displayed when the mouse hovers the button).
Confirmation Message	Displays a confirmation message when the button is clicked. If not informed this is not informed, no message will appear.
CSS Style	Name of the CSS class, style created in the layout editor.
Type	Description of the created button.

Setting up the Link

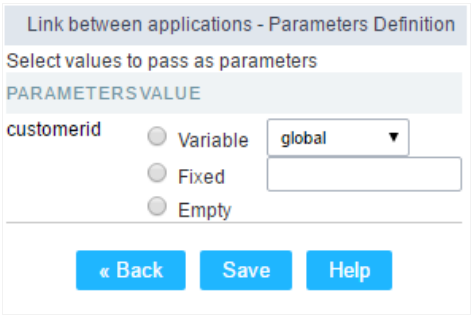
- Selecting the applications



Choosing the application for the button link.

You should select an application to be called from the button link.

- Link Parameters



Choosing the parameters for the button link.

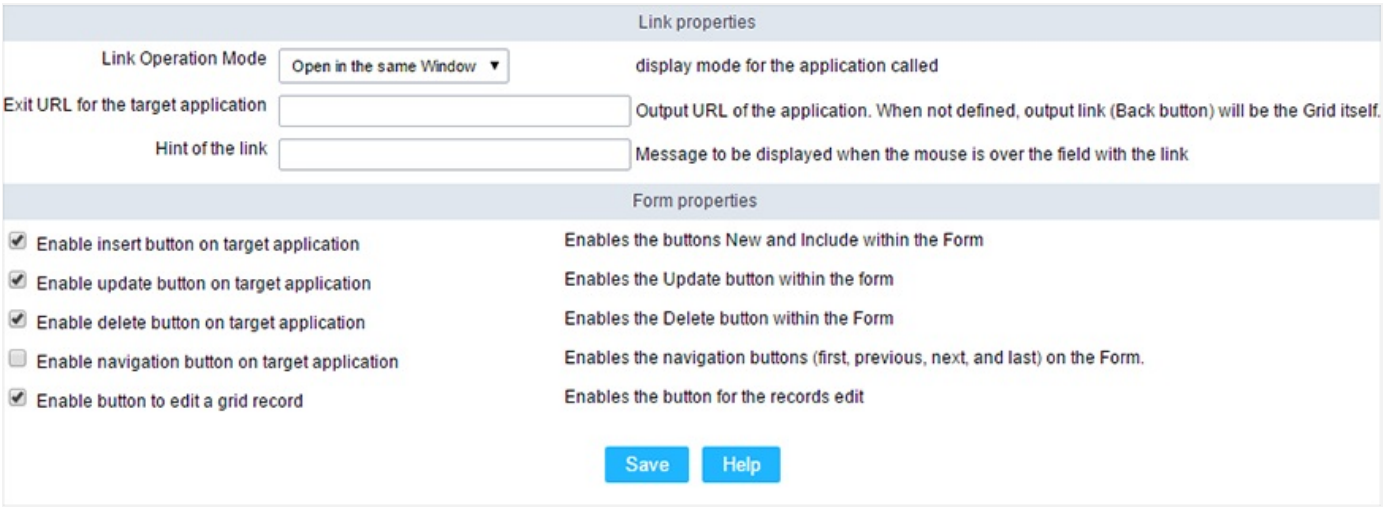
Field Allows you to use an existing field from the current application as a parameter for the link.

Variable Allows you to use a global variable from the current application as a parameter for the link.

Fixed Allows you to inform a fixed value as a parameter for the link.

Empty No value will be passed as a parameter for the link.

- Link Properties (Grid)



Configuring the properties for the link button when the destined application is a Grid.

Link Operation Mode How the link will open (Open in the same Window, Open in another Window, Modal).

Exit URL for the target application URL or an application that be redirect to when exiting the Grid application.

Initial Mode Allows you to define the initial mode of the grid application (Search or Grid).

Number of Lines Allows you to define the amount of lines displayed in the Grid.

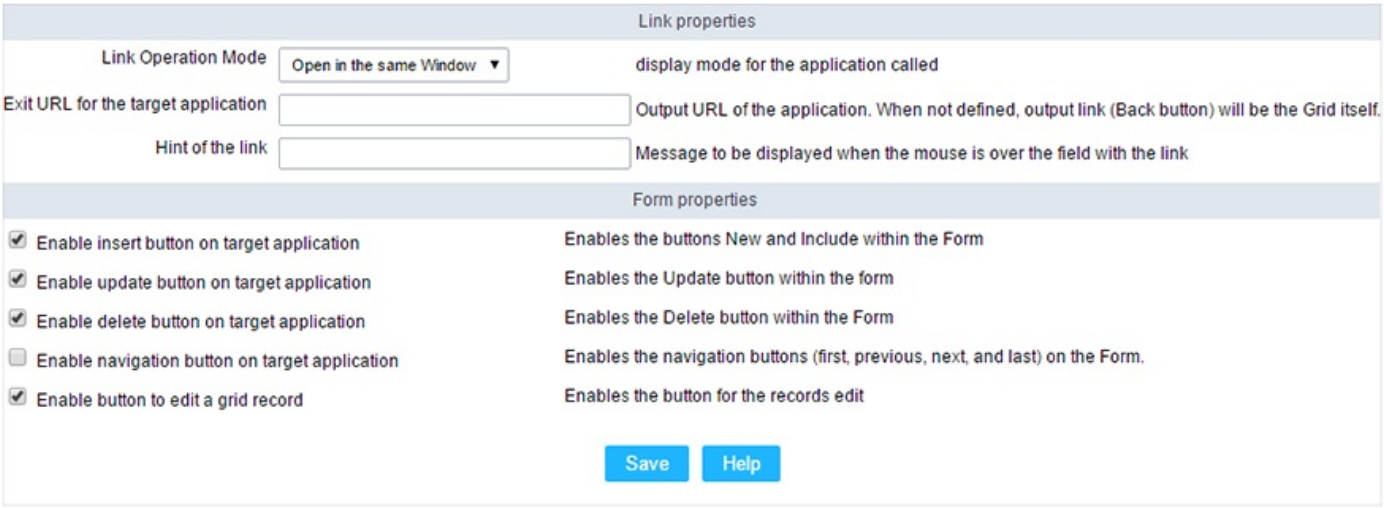
Number of Columns Allows you to define the amount of columns displayed in the Grid.

Paging Enable the paging in the Grid.

Display Header Enable the Grid Header.

Active Navigation Buttons Enable the navigation button (First, Back, Next and Last) in the Grid.

- Link Properties (Form)



Configuring the properties for the link button when the destined application is a Form.

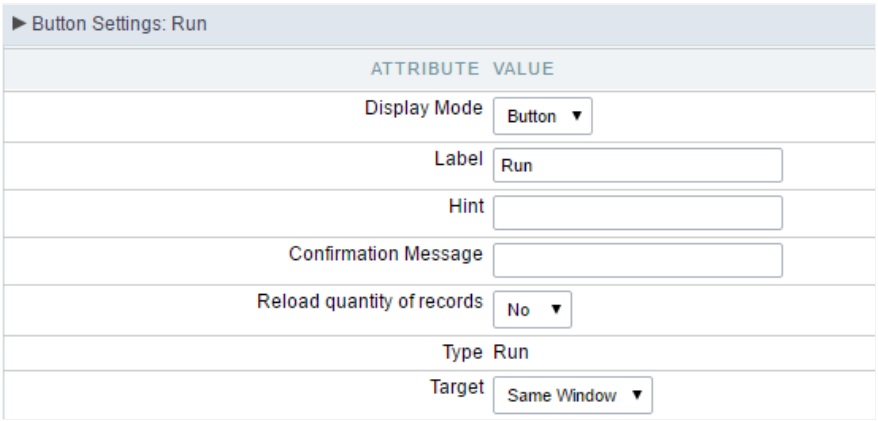
Link Operation Mode	How the link will open (Open in the same Window, Open in another Window, Modal).
Exit URL for the target application	URL or an application that be redirect to when exiting the Form application.
Enable insert button on target application	Enable the “New” button in the Form Application.
Enable update button on target application	Enable the “Update” button in the Form Application.
Enable delete button on target application	Enable the “Delete” button in the Form Application.
Enable navigation button on target application	Enable the navigation button (First, Back, Next and Last) in the Form.
Enable button to edit a grid record	Enable the buttons that allow you to edit the records of a Grid

RUN

Display options

We can configure Run Button display as Link, Image or Button:

Button

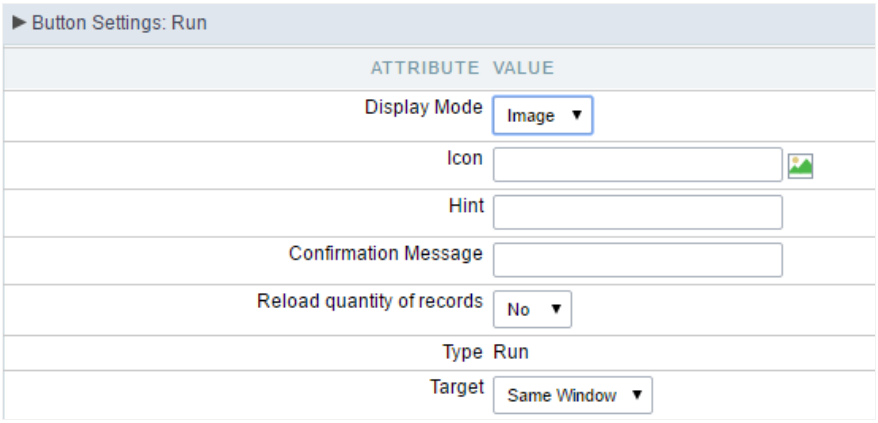


Grid’s Run Button settings - Button type

Display Mode Combo box to select the button display option, you can choose button, image or link.
Label Application button title (text that will be displayed for the button within the application)
Hint Using this option you can set a message for the button hint
Confirmation Message Using this option you can set a confirmation message that will be displayed when the button is pressed. Leave it empty if you don’t need to display a message.

CSS Style	CSS for the the button, if you do not change this option Scriptcase will apply the default application/project CSS. You can customize the buttons' CSS using the option within the main menu "Layout » CSS Buttons"
Reload quantity of records	Option used to update the amount of application records.
Type	Button type description.
Target	This option allows you to set the target window where you will run the button code (the same window, other window, modal)

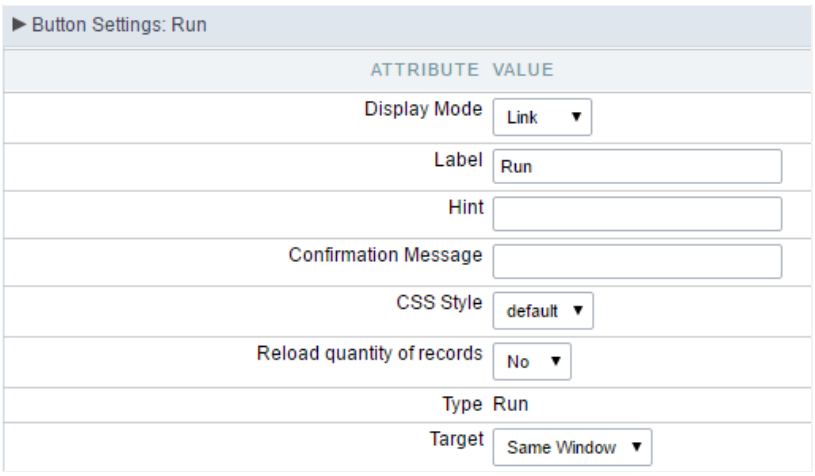
Image



Grid's Run Button settings - Image type

Display Mode	Combo box to select the button display option, you can choose image, button or link.
Icon	Button display icon. You can use this option to select an image (from Scriptcase images or from you computer) to represents the button within the application toolbar.
Hint	Using this option you can set a message for the button hint
Confirmation Message	Using this option you can set a confirmation message that will be displayed when the button is pressed. Leave it empty if you don't need to display a message.
Reload quantity of records	Option used to update the amount of application records.
Tipo	Button type description.
Target	This option allows you to set the target window where you will run the button code (the same window, other window, modal)

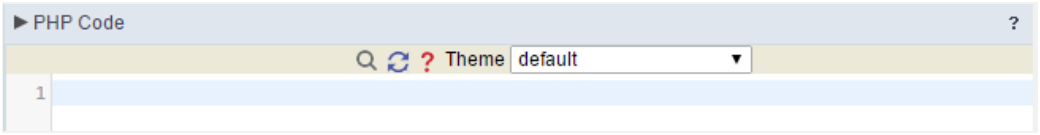
Link



Grid's Run Button settings - Link type

Display Mode	Combo box to select the button display option, you can choose link, button or image.
Label	Text that will be displayed on the button (on running application).
Hint	Application button title (text that will be displayed for the button within the application)
Confirmation Message	Using this option you can set a confirmation message that will be displayed when the button is pressed. Leave it empty if you don't need to display a message.
CSS Style	CSS class name, style created in the theme buttons editor .
Reload quantity of records	Option used to update the amount of application records.
Type	Button type description.
Target	This option allows you to set the target window where you will run the button code (the same window, other window, modal)

Coding Area



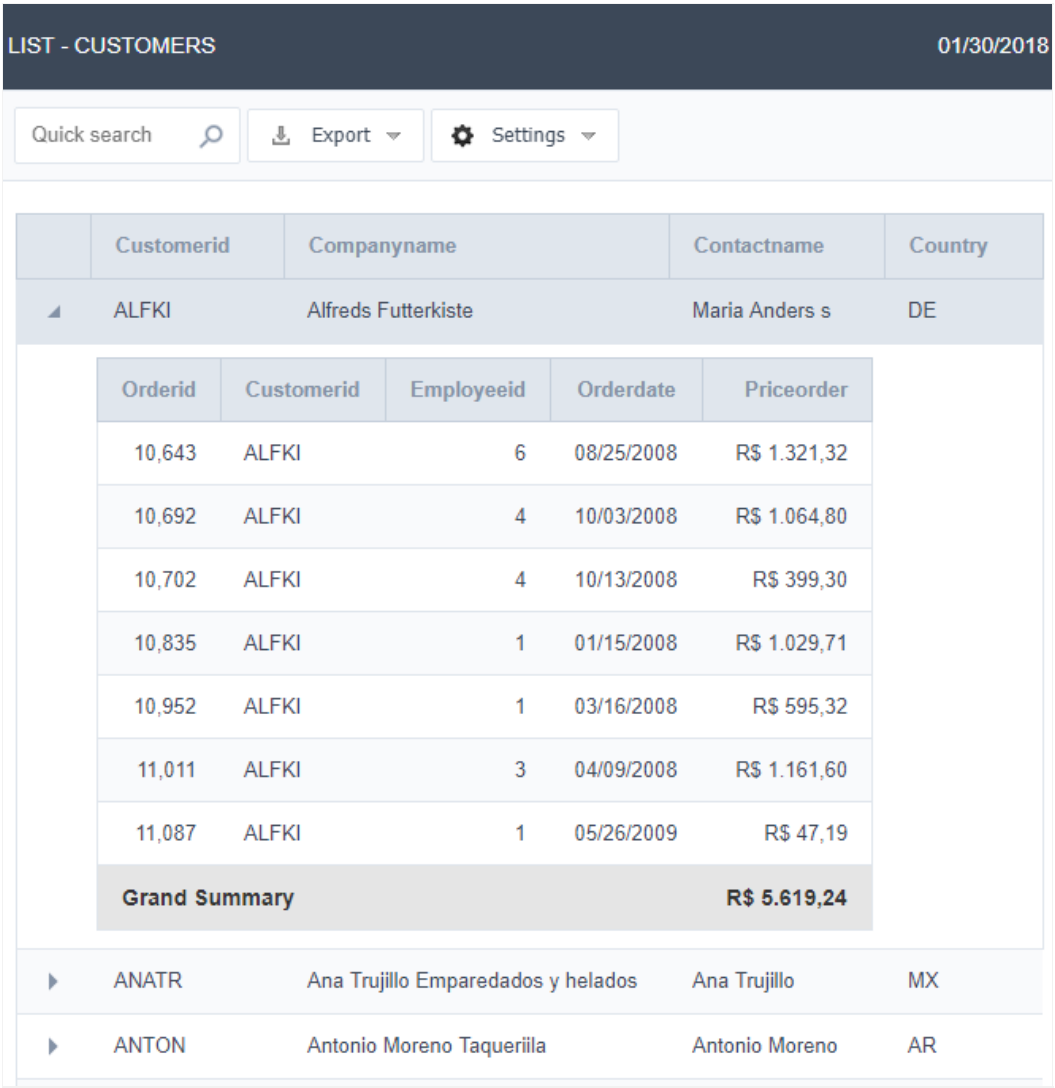
Grid's Run Button settings - Coding area

- There are two types of events in the Run button
  - **OnRecord** : Runs after processing on each record selected.
  - **OnFinish** : Runs after processing all records selected.

In this coding area you can use Scriptcase Macros, PHP and JavaScript.

Procedure Application Nested Grids

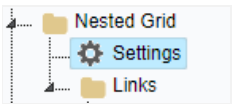
Nested grids are used when you need to display hierarchical data, such as customers and their respective applications. The main Grid encapsulates other Grids, showing in each of its Nested grids additional data records. Note the image below in which for each customer is displayed a sub-query with the applications.



Customer Grid with an order nested grid.

Settings

In the application menu, by clicking on the item folder Configuration Nested grids, can be changed the General attributes of the Nested grids.



Nested grid menu.

Nested Grid Settings

ATTRIBUTE	VALUE	DESCRIPTION
Title in the same line	No	Nested Grid title line will be shown on the same line of the application title.
PDF	Yes	Defines if the Nested Grids will appear in the PDF report
XLS	Yes	It defines if the Nested Grids will appear in the XLS report
XML	Yes	It defines if the Nested Grids will appear in the XML report
Enable TreeView	No	Show the Nested Grid in TreeView mode.
Position	In one column	Position of the Nested Grid.

Nested grid general settings.

- **Attributes**
  - **Title in the same line** : This attribute enables you to configure the Nested Grid title display. If it is displayed on the same line, Nested Grid table will be incorporated into the main Grid.

Customerid	Companyname	Contactname	Orderid	Employeeid	Orderdate	Priceorder
ALFKI	Alfreds Futterkiste	Maria Anders s	10,643	6	08/25/2008	R$ 1.321,32
			10,692	4	10/03/2008	R$ 1.064,80
			10,702	4	10/13/2008	R$ 399,30
			10,835	1	01/15/2008	R$ 1.029,71
			10,952	1	03/16/2008	R$ 595,32
			11,011	3	04/09/2008	R$ 1.161,60
			11,087	1	05/26/2009	R$ 47,19
			Grand Summary			R$ 5.619,24
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	10,248	10	07/04/2008	R$ 232,74

Example using Nested Grid title on the same line.

  - **PDF** : This option allows you to enable/disable the nested grid display in PDF reports.
  - **Enable TreeView** : This option enables an option to hide/display the Nested Grid within the main Grid records. By choosing “Yes”, the option “Title in the same line” will be automatically disabled.

Customerid	Companyname	Contactname
BOLID	Bolido Comidas preparadas	Martin Sommer

Orderid	Employeeid	Orderdate	Priceorder
10,326	4	10/10/2008	R$ 1.195,48
10,801	4	12/29/2008	R$ 4.888,40
10,970	9	03/24/2008	R$ 338,80
Grand Summary			**R$ 6.422,68**

Using tree view in the Nested Grid.

- **Position** : This option allows you to configure the Nested Grid placement in the main Grid. This option is not available if you enable “Title in the same line”. You will be able to position the Nested Grid within the records. The options are:
  - **In one column**

Customerid	Companyname	Contactname	Orders
BLONP	Blondesddsl pere et fils	Frederique Citeaux	
BOLID	Bolido Comidas preparadas	Martin Sommer	

Orderid	Employeeid	Orderdate	Priceorder
10,326	4	10/10/2008	R$ 1.195,48
10,801	4	12/29/2008	R$ 4.888,40
10,970	9	03/24/2008	R$ 338,80
Grand Summary			**R$ 6.422,68**

Customerid	Companyname	Contactname	Orders
BONAP	Bon appart	Laurence Lebihan	
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	

Displaying the Nested Grid in the same row of the main Grid record.

  - **Below the record**

	Customerid	Companyname	Contactname
▲	BOLID	Bolido Comidas preparadas	Martin Sommer

Orderid	Employeeid	Orderdate	Priceorder
10,326	4	10/10/2008	R$ 1.195,48
10,801	4	12/29/2008	R$ 4.888,40
10,970	9	03/24/2008	R$ 338,80
Grand Summary			**R$ 6.422,68**

Displaying the Nested Grid in a separate row of the main Grid record.

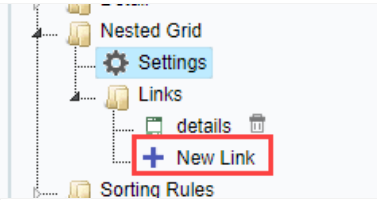
- **Alignment** : Allows you to set the Nested Grid alignment in the main Grid interface when “Position” is set set as Below the record. The options are: left, right and Center.

Set individually

Any of the attributes that use the value “Set individually” requires the property to be configured on each created link settings.

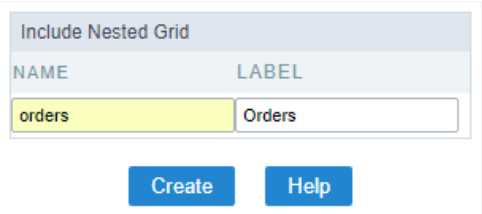
Links

To create a Nested Grid link click on the “New link” within the Link folder.



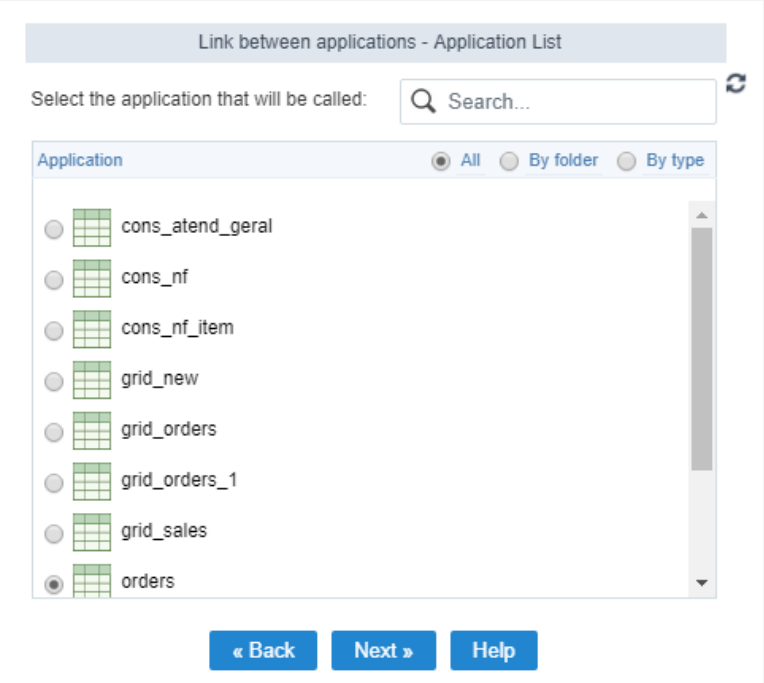
Creating a new Nested Grid

- **Adding a new Nested Grid** : Enter the name and the label for the link that will be created.



Adding a new Nested Grid

- **Application list** : All Grid applications from current project, that contain global variables, will be listed.

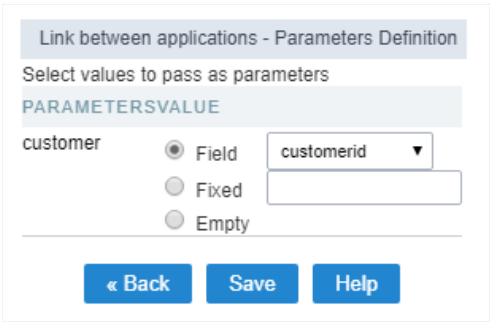


Grid application selection.

- For an application to be used as a Nested Grid, it must have within its SQL command (Grid > SQL) a WHERE clause with a global variable to receive the parameters from the main Grid.

SQL query configuration to the Grid be used as Nested Grid.

- **Parameters setting** : Setting the value that will be passed to the variable(s) of the Nested Grid SQL statement.



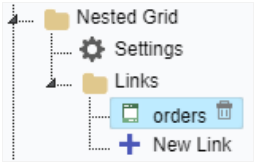
Informing the parameter to the variable

In the screen above, on the left side, it is displayed the Nested Grid input parameters (global variables created within the Nested Grid WHERE clause), on the right side, you must select the parameter option. The options are:

- **Attributes**
  - **Field** : Used to assign a Grid field value as a parameter.
  - **Fix** : Used to assign a fixed value as a parameter.
  - **Empty** : Choosing this option will not be assigned any value as a parameter.
- **Save button** : By clicking this button the Nested Grid creation will be completed and saved.

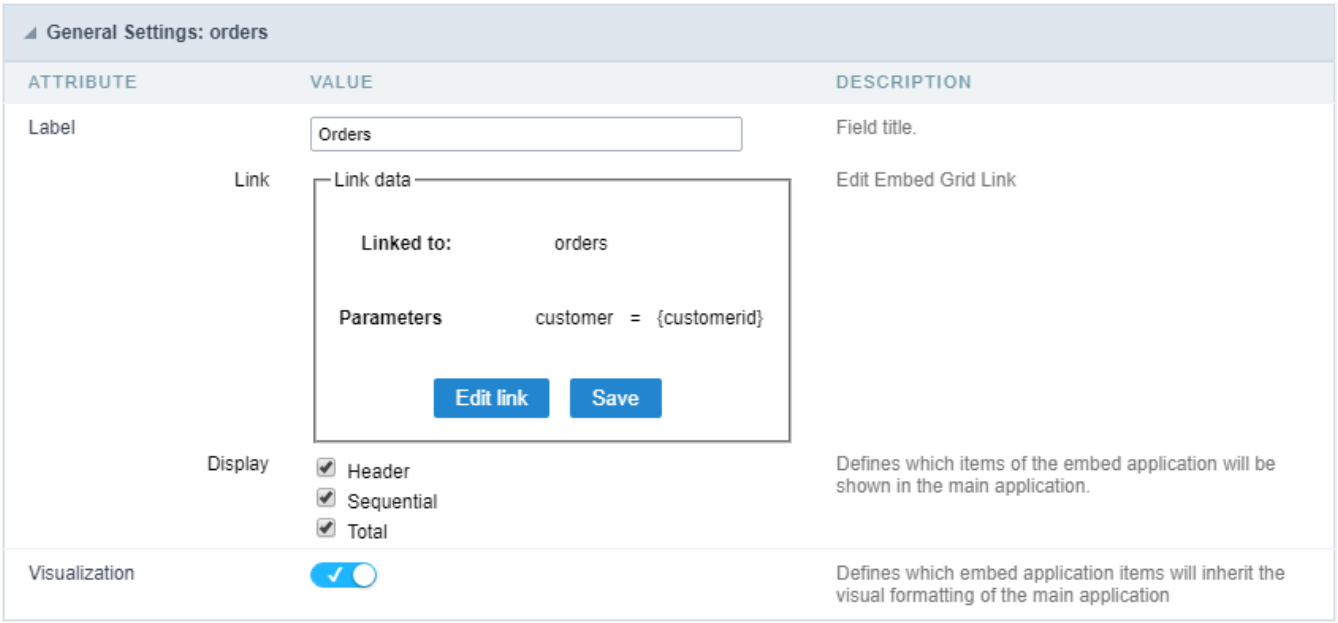
Editing an existing Nested Grid link

All Nested Grid links created will placed within its menu options, inside the folder “Links”. By clicking on the Nested Grid name there will be an interface for settings editing.



Editing a Nested Grid

General settings



Nested Grid general settings

This interface can be used to edit the following Nested Grid attributes:

- **Label** : Nested Grid title for the application
- **Link** : In this attribute are displayed the current Nested Grid connection data, showing the application and parameters used. To change any link attribute just click to edit Link.
- **Display** : This attribute allows you to set which Nested Grid items, if it uses any of these resources, will be displayed in the main Grid.

	Orderid	Employeeid	Orderdate	Priceorder
1	10,326	4	10/10/2008	R$ 1.195,48
2	10,801	4	12/29/2008	R$ 4.888,40
3	10,970	9	03/24/2008	R$ 338,80
Grand Summary				**R$ 6.422,68**

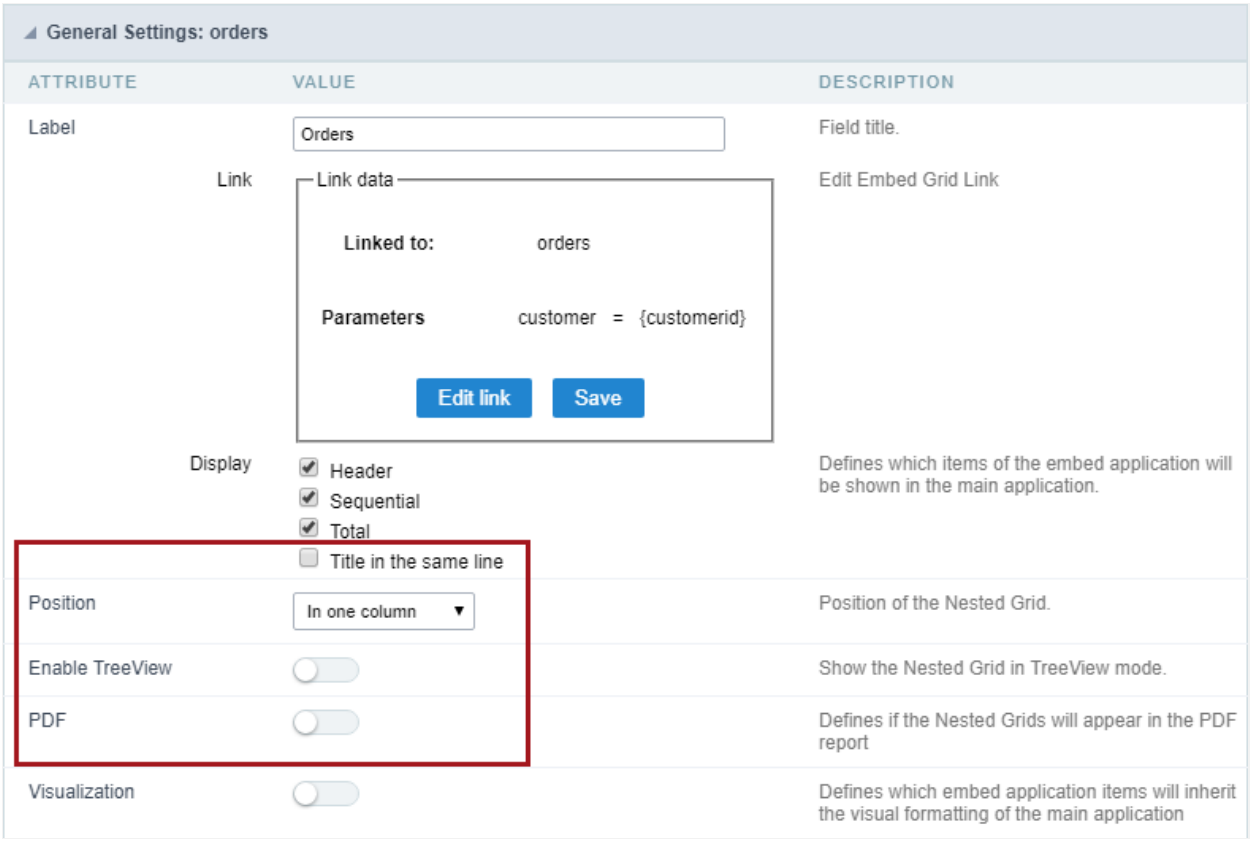
Display settings

* __1__ : Display Title.
* __2__ : Sequential display (number line).
* __3__ : Display Totals.

- **Inherit view** : When marked it sets that the Nested Grid view will be equal to that of the main Grid.

Set individually

If the value of one of the attributes in the Nested Grid settings has been selected as “Set Individually”, these values must be configured in the Nested Grid editing screen for each called link.

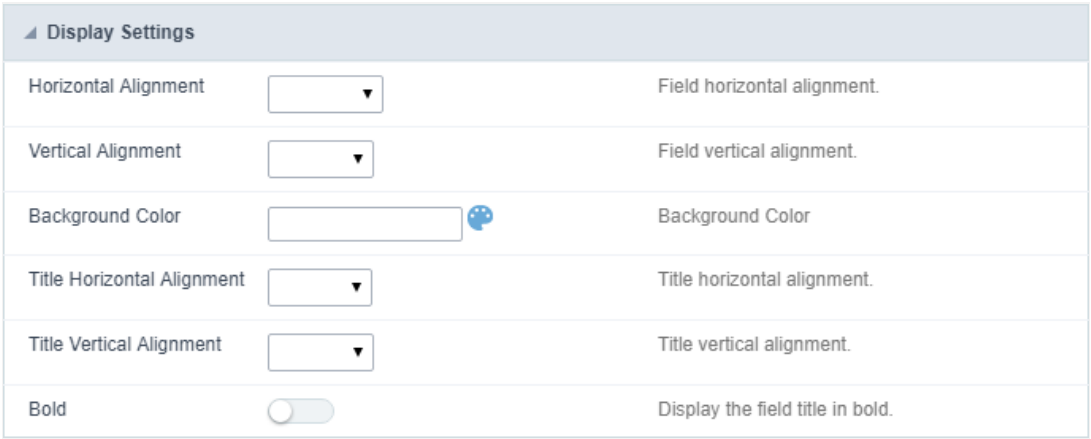


Nested Grid settings.

Display settings

These options will only appear if all of the following conditions are true:

Attribute	Value
Title in the same line	No
Enable Tree view	No
Position	In one column



Nested Grid display settings

- **Attributes**
  - **Horizontal Alignment** : This option sets the Nested Grid horizontal alignment. It can be aligned left, right or Center.
  - **Vertical Alignment** : This option sets the Nested Grid vertical alignment. It can be aligned to the top, middle, or End.
  - **Background color** : Here you can set the background color. If it is not filled it will get the Grid default background color.
  - **Title horizontal alignment** : This option sets the Nested Grid label horizontal alignment. It can be aligned Left, Right or

Center.

- **Title vertical alignment** : This option sets the Nested Grid label vertical alignment. It can be aligned to the top, middle, or End.
- **Bold** : This option sets the Nested Grid title (label) in bold.

Procedure Application totals

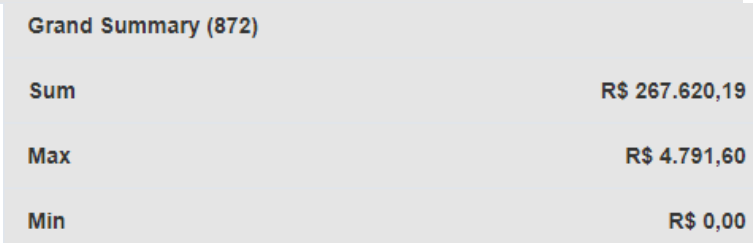
The total fields will be displayed only when the Grid application is using at least one Group By.

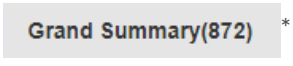
Settings

The settings below only can be applied for the Grid totals.

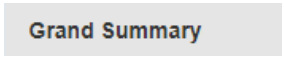
- **Results in a single line.**: This option sets the display of the **General Total** title and its results in a single line.
  - Example for this option **Enabled**:

  - Example for this option **Disabled**:



- **Display Total** : This option indicates in which pages the General Total will be displayed. The options are: **On every page**, **On the last page** or **Do not display**.
- **Group Subtotal** : This option defines where the subtotal group by will be displayed.
- **Record Count** : This option allows you to view the amount of records by the General Total title.
  - Example for this option **Enabled**: 

* Example for this option **Disabled**:



Layout settings

Setting of display of the subtotal group.

- **Label** - This field sets the Label for the **Group Subtotal**.
- **Display** - This option sets if the **Group Subtotal** it will be display.

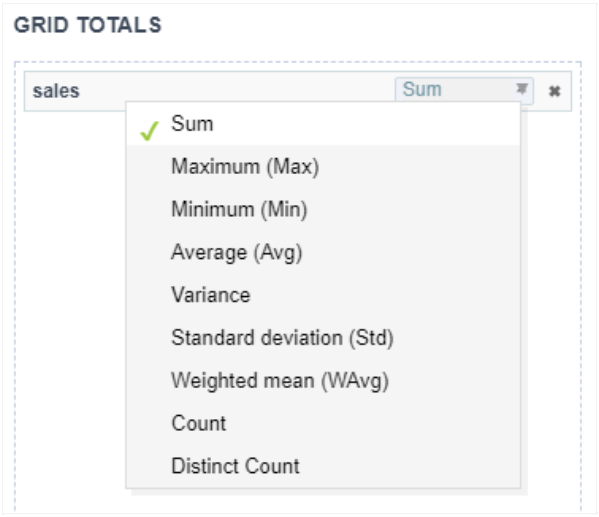
Select fields

Using the select fields area you can set the total fields ant the total options for each field. The same field can be used more than once in the Grid totals area, just if they are using different summaries types.

To define the fields that will be used for totals, drag them to the area, **Grid Totals**.

Note: The total fields will only be displayed if they are also selected to be displayed within the Grid application module.

When positioning the fields, you must define what summarization will be used, to select that you must click in the combo box and select one of the available options, that will be according to the data type (integer, date, text...).



The summarization options available are:

- **Sum** : Sets a sum of the values for the selected field.
- **Maximum** : Displays the highest value identified in the selected field.
- **Minimum** : Displays the lowest value identified in the selected field.
- **Average (Avg)** : Calculates the arithmetic mean of the values for the selected field.
- **Variance** : Calculates the dispersion of the values related to the average.
- **Standard Deviation** : Measures the variability of values around the average, the minimum value of the standard deviation is 0 indicating that there is variability, i.e. that all values are equal to the mean.
- **Weighted mean (WAvg)** : Calculates the weighted average for the selected field. To set the weight used in the calculation of the average access field settings selected in **Totals > Fields (select the field where you are using the Weighted mean) > Weighted average weight**.
    - **Weighted average weight** : Field that will be used as weighted average weight. In calculating the weighted average, each set value is multiplied by its “weight”, that is its relative importance.
- **Count** : Displays the total number of records for the selected field.
- **Distinct Count** : Displays the total number of records for the selected field, distinguishing the values.

Important note: The fields in the Grid totals are displayed only on Grids with no Group by or with empty Group by.

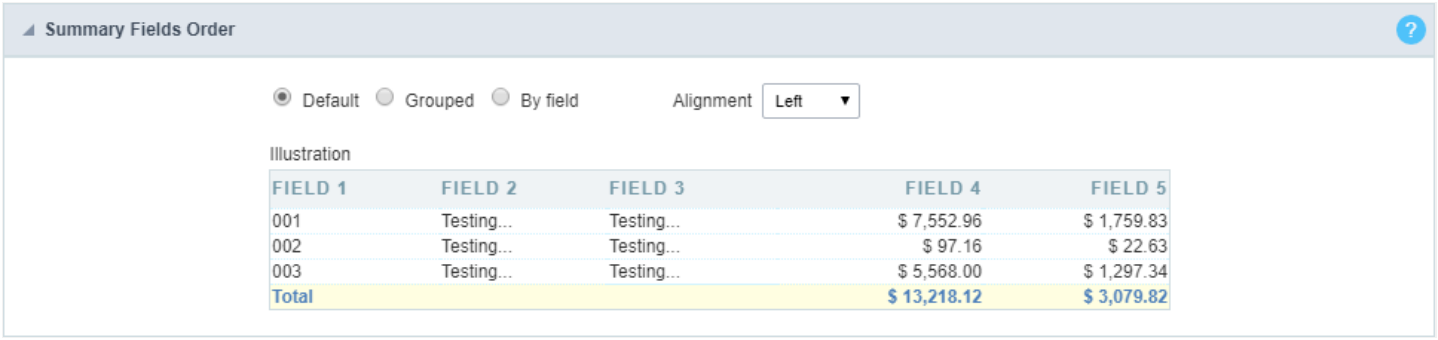
Positioning

Defines the positioning and the label used by Grid totals. There are three display formats,, **Default**, **Grouped** or **By field**

Default

To add the same field two or more times in the Totals or use different types of summaries in selected fields, this option will not be displayed.

The option **Default** returns the result below the column being summarized. When you use the total for more than one column of the Grid, using the same type of summarization, the results are displayed on the same line.



The line with **General Total** displaying the **Sales**.

Id	Date	Region	Product Category	Sales
1	01/01/2015	South Atlantic	SEAFOOD	R$ 50,82
2	01/01/2016	South Atlantic	CONFECTIONS	R$ 84,70
3	01/01/2013	South Atlantic	BEVERAGES	R$ 90,75
4	01/02/2013	East South Central	CONFECTIONS	R$ 163,35
5	01/02/2014	East South Central	DAIRY PRODUCTS	R$ 251,68
Grand Summary				**R$ 267.620,19**

Alignment

This option is available for display formats **Default** and **Grouped**.

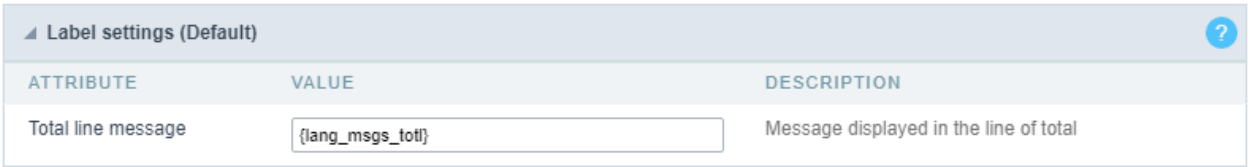
Defines the placement of the label selected within the **Setting > Display total** option. The alignment can be in the **Center**, **Left** e **Right**:

- **Left** : **Grand Summary** **R$ 267.620,19**
- **Right** : **Grand Summary** **R$ 267.620,19**
- **Center** : **Grand Summary** **R$ 267.620,19**

Label settings (Default)

This option allow you to customize the total labels.

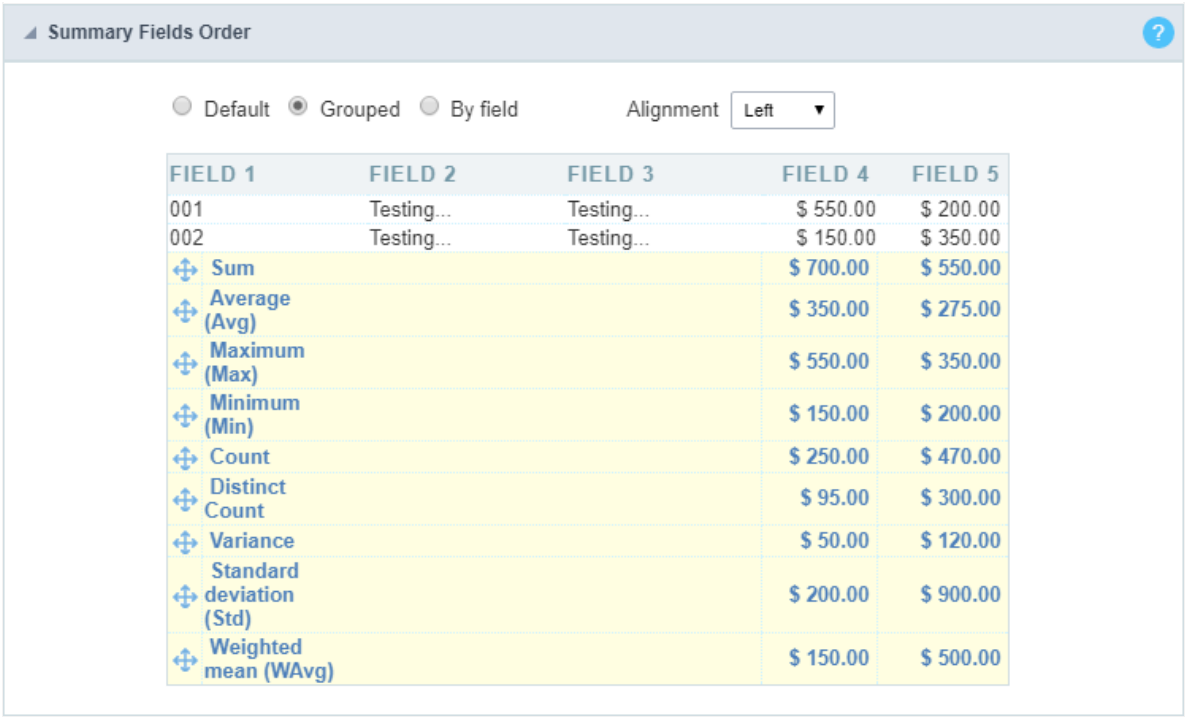
As default we are going to display Grant total



Grouped

The option **Grouped** returns the total result, separating each type per line.

It is possible to position the total lines by dragging to the desired position.



In this example, we are displaying the sum, average, maximum and minimum of column **Sales**.

Id	Date	Region	Product Category	Sales
1	01/01/2015	South Atlantic	SEAFOOD	R$ 50,82
2	01/01/2016	South Atlantic	CONFECTIONS	R$ 84,70
3	01/01/2013	South Atlantic	BEVERAGES	R$ 90,75
4	01/02/2013	East South Central	CONFECTIONS	R$ 163,35
5	01/02/2014	East South Central	DAIRY PRODUCTS	R$ 251,68
Grand Summary				
Sum				**R$ 267.620,19**
Avg				**R$ 306,90**
Max				**R$ 4.791,60**
Min				**R$ 0,00**

Alignment

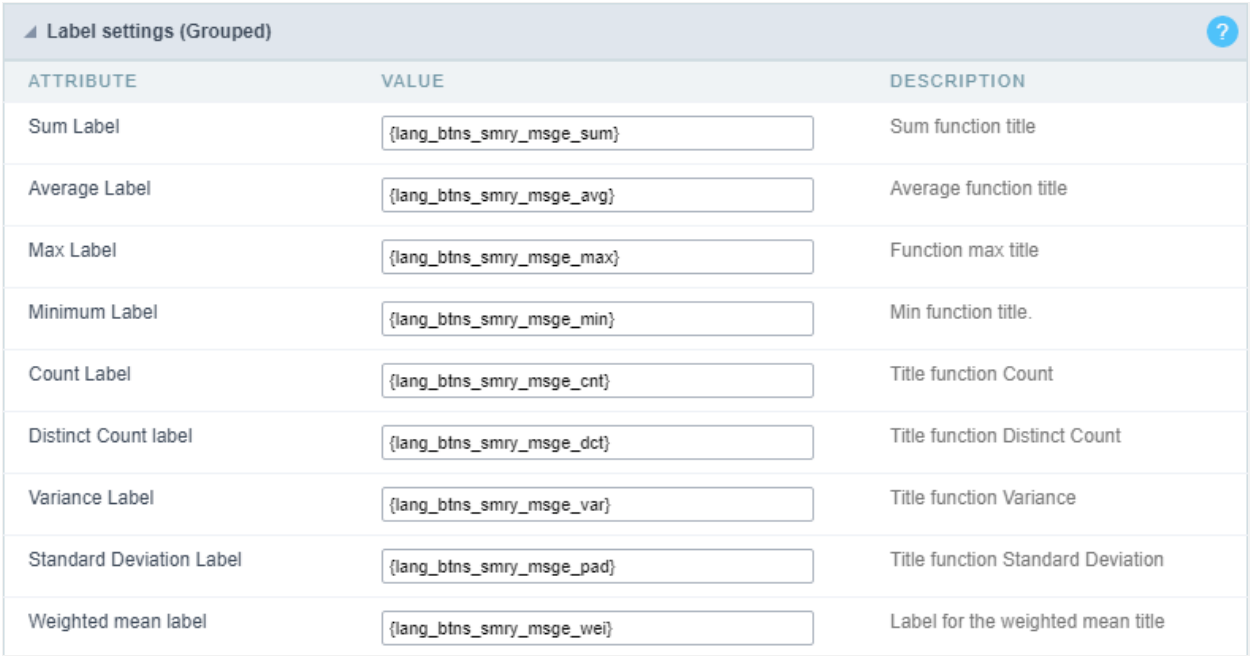
This option is available for display formats **Default** and **Grouped**.

Defines the placement of the label selected within the **Setting > Display total** option. The alignment can be in the **Center**, **Left** e **Right**:

- **Left** : **Grand Summary** **R$ 267.620,19**
- **Right** : **Grand Summary** **R$ 267.620,19**
- **Center** : **Grand Summary** **R$ 267.620,19**

Label settings (Grouped)

This option allow you to customize the total labels.

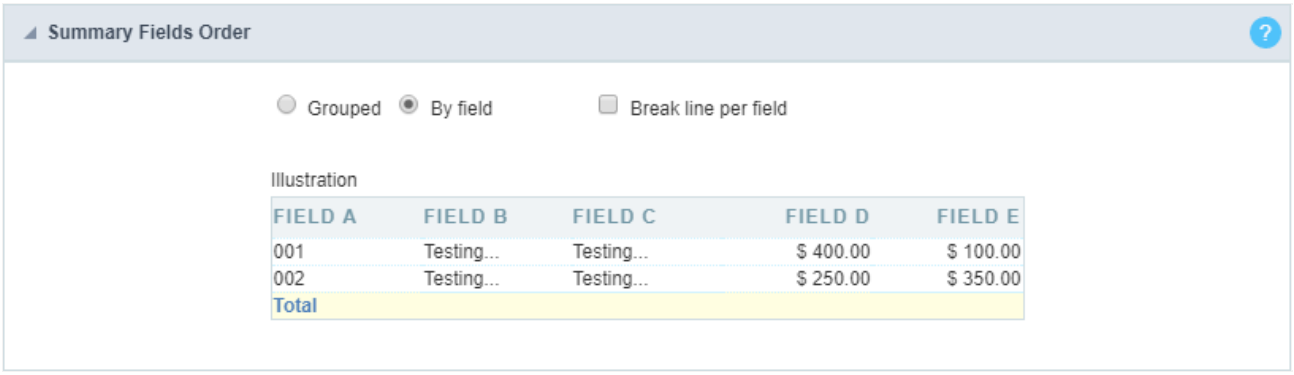


Label settings (Grouped)

ATTRIBUTE	VALUE	DESCRIPTION
Sum Label	{lang_btns_smry_msge_sum}	Sum function title
Average Label	{lang_btns_smry_msge_avg}	Average function title
Max Label	{lang_btns_smry_msge_max}	Function max title
Minimum Label	{lang_btns_smry_msge_min}	Min function title.
Count Label	{lang_btns_smry_msge_cnt}	Title function Count
Distinct Count label	{lang_btns_smry_msge_dct}	Title function Distinct Count
Variance Label	{lang_btns_smry_msge_var}	Title function Variance
Standard Deviation Label	{lang_btns_smry_msge_pad}	Title function Standard Deviation
Weighted mean label	{lang_btns_smry_msge_wei}	Label for the weighted mean title

By field

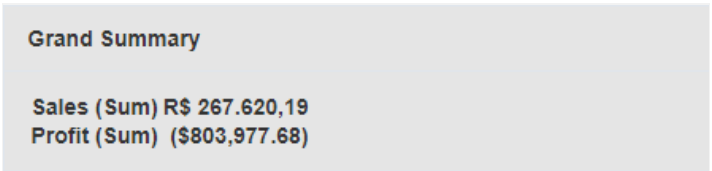
This option displays the values in the left corner by positioning the results next to each other.



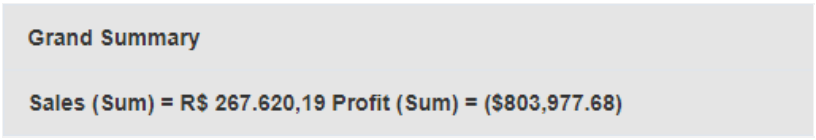
Line break per field

This option sets the total fields display. If selected it shows the fields in the same row or divided by line.

Separated by line:



Displayed on the same line:



Label settings (By field)

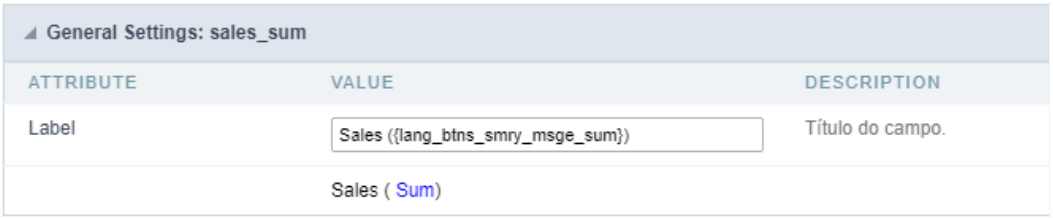
This option allow you to customize the total labels.

Fields

This option allows you to configure the display of values and labels of the selected fields in the Grid totals.

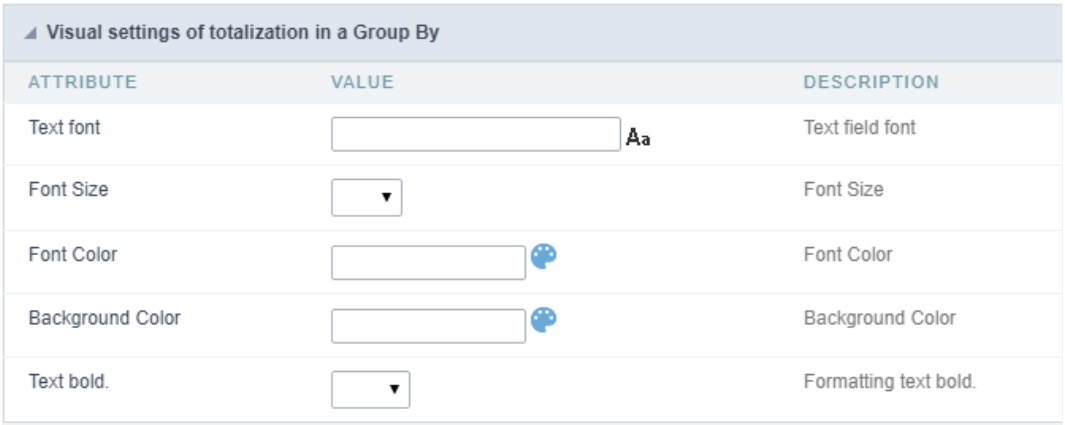
General Settings

Allows you to change the label displayed in the totals



Visual configuration for the Grid totals

Sets the formatting of the fields displayed in the subtotal for the group.



- **Font family** : Sets the font used.
- **Font size** : Sets the font size.
- **Text color** : Sets the text color
- **Background color** : Sets the background color.
- **Bold** : Format text in bold.

Example of formatting the subtotal for the Group:

Date of Date => 01/09/2013

Id	Date	Region	Product Category	Sales	Profit
22	01/09/2013	East South Central	BEVERAGES	R$ 163,35	$36.30
19	01/09/2013	East South Central	GRAINS/CEREALS	R$ 363,00	$121.00
25	01/09/2013	East North Central	CONFECTIONS	R$ 116,16	$19.36
Total sales on the day - 01/09/2013				R$ 642,51	

General Total Visual setting

Sets the formatting of the fields displayed in the General total.

Visual setting of the General Total

ATTRIBUTE	VALUE	DESCRIPTION
Text Font	Aa	Field Text size
Font Size		Font Size
Font Color.		Font Color.
Background Color		Background Color
Bold text		Formating text in bold.

- **Font family** : Sets the font used.
- **Font size** : Sets the font size.
- **Text color** : Sets the text color
- **Background color** : Sets the background color.
- **Bold** : Format text in bold.

Example of formatting the Grid General Total

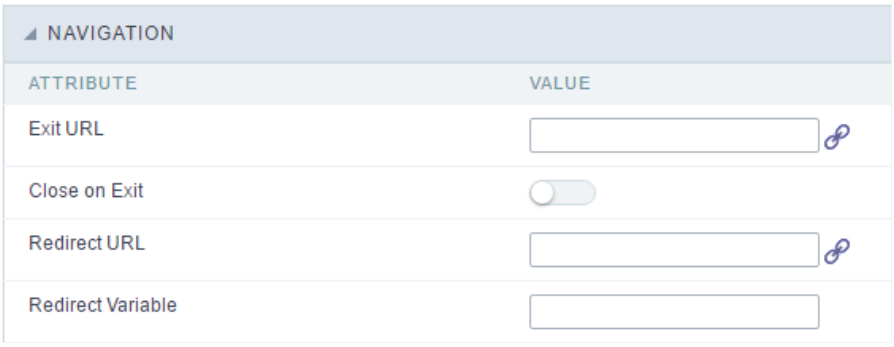
Id	Date	Region	Product Category	Sales	Profit
1	01/01/2015	South Atlantic	SEAFOOD	R$ 50,82	$14.52
2	01/01/2016	South Atlantic	CONFECTIONS	R$ 84,70	$33.88
3	01/01/2013	South Atlantic	BEVERAGES	R$ 90,75	$36.30
4	01/02/2013	East South Central	CONFECTIONS	R$ 163,35	$54.45
5	01/02/2014	East South Central	DAIRY PRODUCTS	R$ 251,68	$94.38
Grand Summary				R$ 267.620,19	

Application Settings (Procedure)

Settings

Navigation

This interface allows defining the navigating behavior of the application



Navigation Interface.

Exit URL

URL to where the user goes when he clicks on the “exit” button.

Close on Exit

Close the browser window when the user clicks on the “exit” button.

Redirect URL

Redirect to another URL in case there aren’t any global variables available.

Redirect Variable

Creates a variable with the application name and sends it to the redirected application.

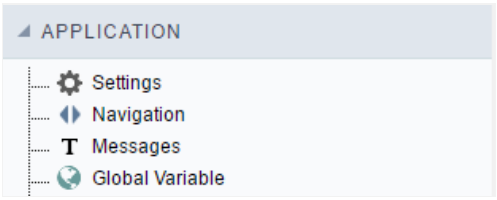
Global Variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

IMPORTANT: if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**“**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target=”_blank”} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.



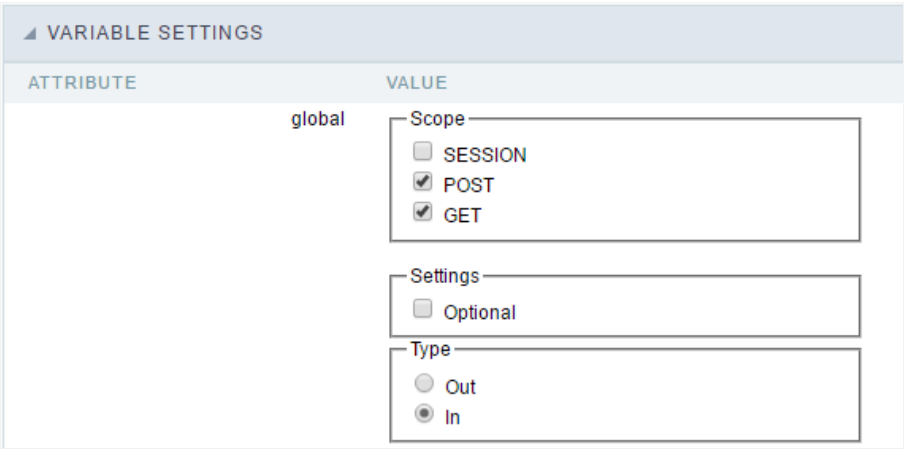
Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.



Global variables configuration Interface.

Attribute

It shows the variable's name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don't validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

Indicates where the application is using the variable.

Procedure Links Overview

This feature allows the developer to create links between applications of the same project, expanding the integration of applications. All link options are grouped under the Application Links menu.

Application Links

In the first access to the menu, we can see the list of existing connections in the application.

> If the application does not have a configured link, the application list screen will be displayed with the message: **This application does not have any link. Click here to create one now.**

ID

Link identification ID.

Type

The type of link created, some links such as Edit Link allow only one link per application. In this case, the developer will be able to check the connection types that already exist in the application

Target Application

This info show the target application name.

Actions

This column has edit options for the links.

Properties

It allows accessing the binding properties where it is possible to configure the binding behavior.

Link

Displays the links screen, where it is possible to configure the link that was made with the application informed in the target application column. In this option it is possible to change the parameter passed in the connection as well as the target application.

Delete

Permanently deletes the connection in the applications.

> Through the option Restore Applications, it is possible to get a previous version of the application, making it possible to recover the link deleted.

Links Type

The Procedure application has the following links options

- Application Link - Allows the creation of a link with Procedure application in order to edit the records listed in the Grid.
- Capture Link: It allows the creation of links from the filter fields of the Grid application, in order to enable the recovery of the value to fill in the field, with another Grid application of the project.
- Field Link: Allows you to create a link through the application fields to any application in the project.
- Button Link: It allows creating a link through the buttons created by the developer in the query application, to any application in the project.

Grid Applications Link Types x Application available to link

Check the list of applications that can be associated with the available connection types.

The restrictions on the choice of applications are the **Edit Link** which only allows linking with a form and the **Capture Link** where it is possible to link the filter fields with only one other Procedure application.

	Application Link	Field Link	Capture Link	Button Link
Grid				
Chart				
Form				
Control				
Search				
Menu				
Tree Menu				
Tabs				
PDF Report				
Dashboard				
Blank				
Calendar				

Related Link

- Drill Down
- Master/Detail on 3 levels
- Report link to edit the record
- Modal Links
- Macro sc_link

Related Video

New link

![Links menu interface.][ligacao_nova] *Links menu interface.*

When you click the “New Link” item from ScriptCase application menu will, the screen below will be displayed.

![Interface for creating links][ligacao_criando] *Interface for creating links*

Application Link

Used to edit records of a Grid Application by using a Form Application. In the generated Grid you will view a link for each record to edit the records. Clicking on the link, the form selected can be displayed in various ways (in an iframe, in the same window or on another window).

List of Applications

> Application: You need to select the form application that is going to be called by the Grid Application.

Parameters Definition

Parameters Definition Interface.

On the screen above, on the left side, are displayed the parameters that are expected for the form application (Primary Key, Global Variables), on the right side, you need to select the option that is going to be set to the parameter. The options are:

- **Field** : Used to pass a field value of a Grid as a parameter.
- **Value** : Used to pass a static value as a parameter.
- **Variable** : Used to pass a global variable value used in the Grid as a parameter.
- **Empty** : Choosing this option, no value will be passed as a parameter.

Fiels Link

Capture Link

Creating a Capture Link

The capture link is used to return a value from a **Grid** to a **Form** field.

In the types of links options, we will choose the **Capture Link**. After selection this option, we should also choose which field we want to return the value of.

Applications list

After we select this option, will be displayed the applications list to what you want to create the link.

This screen can be viewed from the following ways:

All:

In which is possible to see all the project's applications. **Example**:

By folder:

In which is possible to see the applications according to the folders in which each of them are.

By type:

In which is possible to see the applications grouped by its respective types.

Definition of Parameters

In this setting we are going to define the type of parameter that will be passed to the next application.

We have two options at **Type of Parameters**, they are:

Fixed value:

This option allows the user to define a fixed value that will be used to the call of the next application.

No value:

This option allows to create the link without the need to send any parameter to the next application.

It is also possible to see a refresh button at its right side.

This button should be used when a new parameter is added to the target application, so the new parameter can be loaded in the current application to make the link.

Application with no parameters

However, when the target application does not have any defined parameters, the following screen is displayed:

Clicking in the button, you will be taken to the target application to create a parameter, so you can use the update button in the **parameters definitions** to refresh them.

Link properties

In this screen we will set the application display mode that will be called in the link.

In this type of link there is only one display option:

Modal:

When we use this option the target application will be displayed in the same window of our application, however it will be displayed in a small window above the current application.

Modal

When we use this option the target application will be displayed in the same window of the application, however she will be displayed in a small window above the current application.

When we select this option, we can set the following options:

Modal:

In this option we are going to define if the target application will be opened in a Modal.

Yes: This option will make the target application be opened in a modal. **No**: This option will make the target application be opened in a new window.

If **Yes** is selected in the previous option, the Modal **Height** and **Width** will be available.

![Modal with yes][modal2]

Height:

Allows to set the Modal height.

Width:

Allows to set the Modal width.

If **No** is selected in the previous option, only those options will be available.

Allows to modify manually in the update:

When we select the option **Yes**, the button **New** is created in the grid to insert new registers.

Allows to modify manually in the insert:

Indicates which shortcut key to the button add new register.

Allows to modify automatically in the update:

Exit URL to the called form. If any value is defined, the return address (back button) will be the exit URL of the called form.

Allows to modify automatically in the insert:

Choosing the option **Yes**, the window will be closed after doing the insert of a register.

Button Link

Links Edit

In the Link Folder of the Application Menu (Image Below) are displayed the links existing in the application and also the item New Link. When clicking on the existing link it is displayed the screen below that allows to manage the links.

Editing Links.

Actions

Properties	Change the link’s behavior, position, and how the link opens.
Link	Change the application that’s being called in the link their parameters.
Delete	Remove the existing link.

Link Propeties

On the screen below, you can see the attributes relevant to the link behavior of that need to be informed.

Link properties Interface.

- **Link Operation Mode** :
  - **Open in the same Window** : Displays the Form in the same browser window.
  - **Open in another Window** : Displays the Form in another window of the browser.
  - **Open in Iframe** : The Form will be displayed in the same window, allowing to position below, above, to the left or right of the Grid.
  - **Open in Parent** : If the Grid is in a iframe, it displays the Form in the page's parent view port.
  - **Modal** : Opens the Form in modal, allowing to configure the modal size.
- **Display button (new row) on the grid toolbar** : This option, when enabled, adds a button to the toolbar that allows to add a New Record to the Form.
- **Label for the button New** : You can apply a Label for the button, if not informed the label will be "New".
- **Hint for the button New** : You can apply a Hint for the button, if not informed the hint will be "New".
- **Hotkey for the button "New"** : Indicates which key will be the shortcut for the New Record.

- **Exit URL for the target application** : The URL that is going to call after exiting the Form. In case no value is informed, the "back" button will redirect to the Grid.

- **Form Property**

  Allows to select the buttons that are going to be displayed in the Form through the link.
  Form properties configurations Interface.

  - **Enable insert button on target application** : Enables the Insert Button in the Form.
  - **Enable update button on target application** : Enables the Update Button in the Form.
  - **Enable delete button on target application** : Enables the Delete Button in the Form.
  - **Enable navigation button on target application** : Enables the navigation buttons (first, previous, next and last) in the Form.
  - **Enable button to edit a grid record** : Enables the edit button for each record.
- **Iframe properties**

  These options are displayed when the property Link Operation Mode is set to "Open in Iframe".
  Iframe properties configurations Interface.

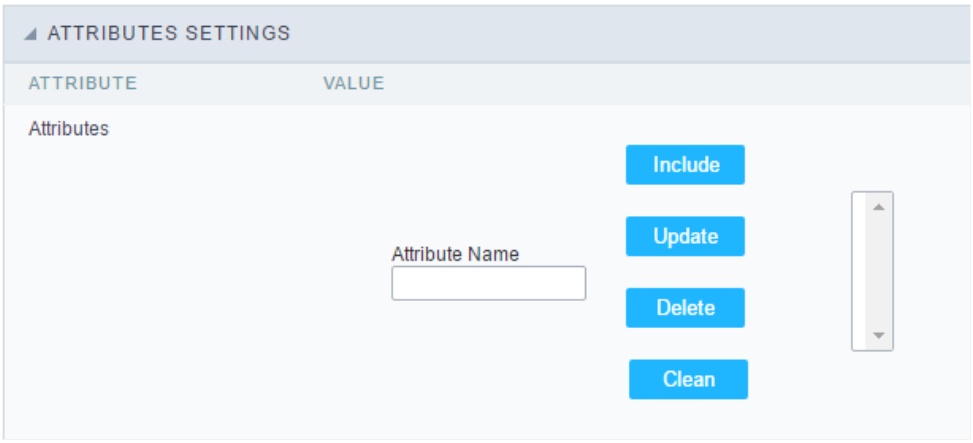
  - **Display the header of the called application** : Displays the Form's header.
  - **Iframe position relative to the main application** : It can be: below, above, right or left.
  - **Action After Insert** view the list below :
    - **Reload Grid** : Does a refresh on the current page.
    - **Move to the end of the grid** : Navigate to the Grid's last page.
  - **Iframe height** : Iframe Height in pixels.
  - **Iframe width** : Iframe Width in pixels.

Procedure Programming

The concept of programming is incorporated in this ScriptCase version. With the use of resources of attributes, methods and libraries. In the previous version was already possible create business rules in the applications, using this concept. The biggest difference is that now this can be realized in a more organized way, making it easier to be developed and comprehended by another developer.

Attributes

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application's events.



Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons' right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal Libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application's Events and Methods.

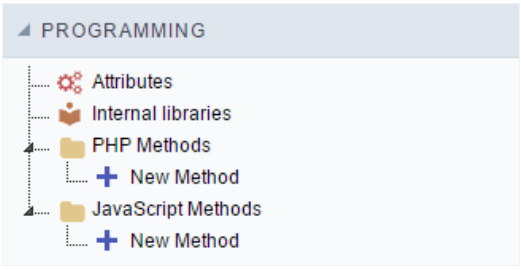
See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

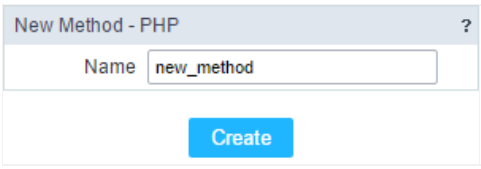
Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

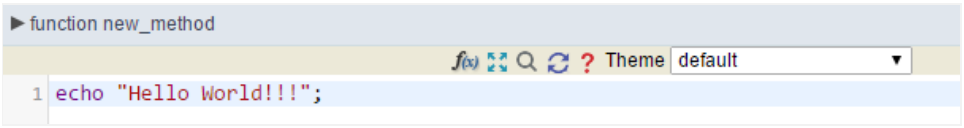


- Give the method a name and click Create. As in the image below.

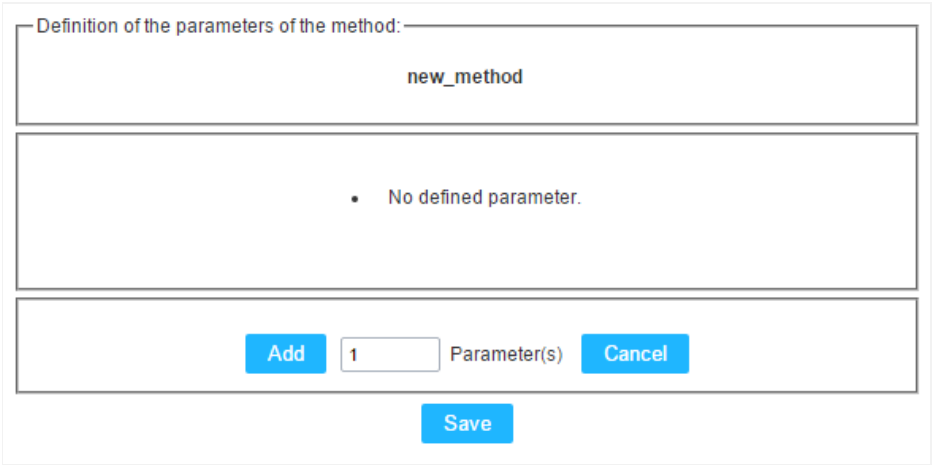
PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.



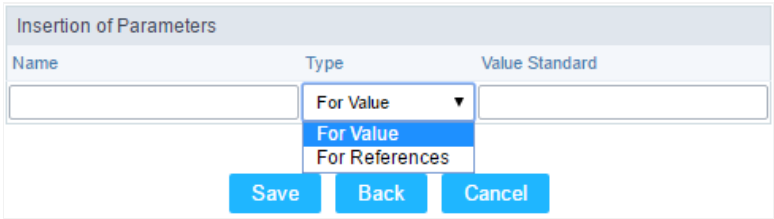
- Methods can receive parameters.



- Add the amount of variables:



- Defining the variables:

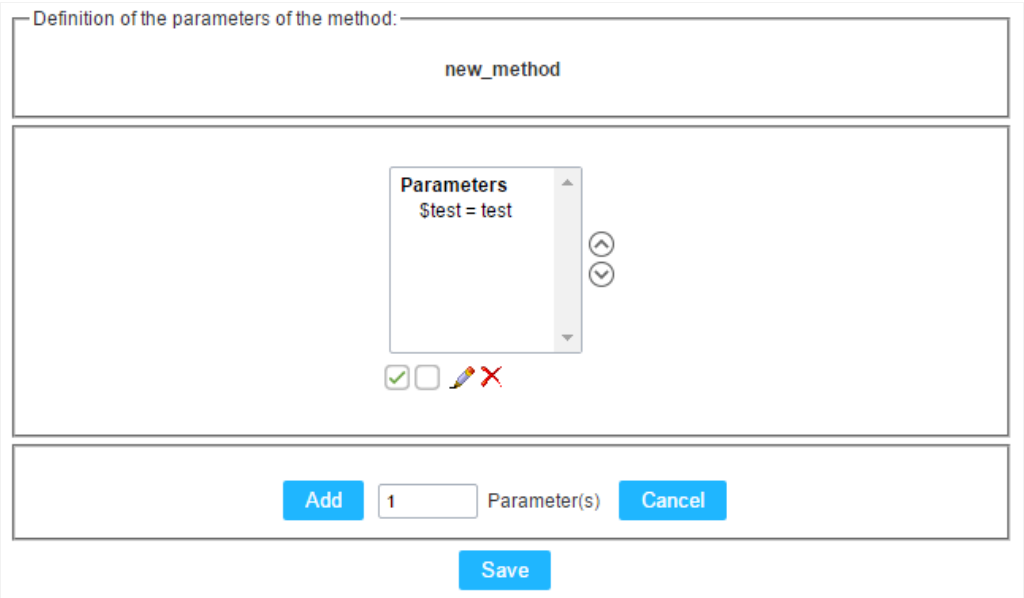


- **Name** : Type in the variable's name.

- **Type** : Selecting the type of variables: For Value or For Reference.
- **Value Standard** : The parameter's value used to initialize when calling the method.

References allows to create a second name for a variable that you can use to read and modify the variable's original information.

- Editing a parameter:



- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

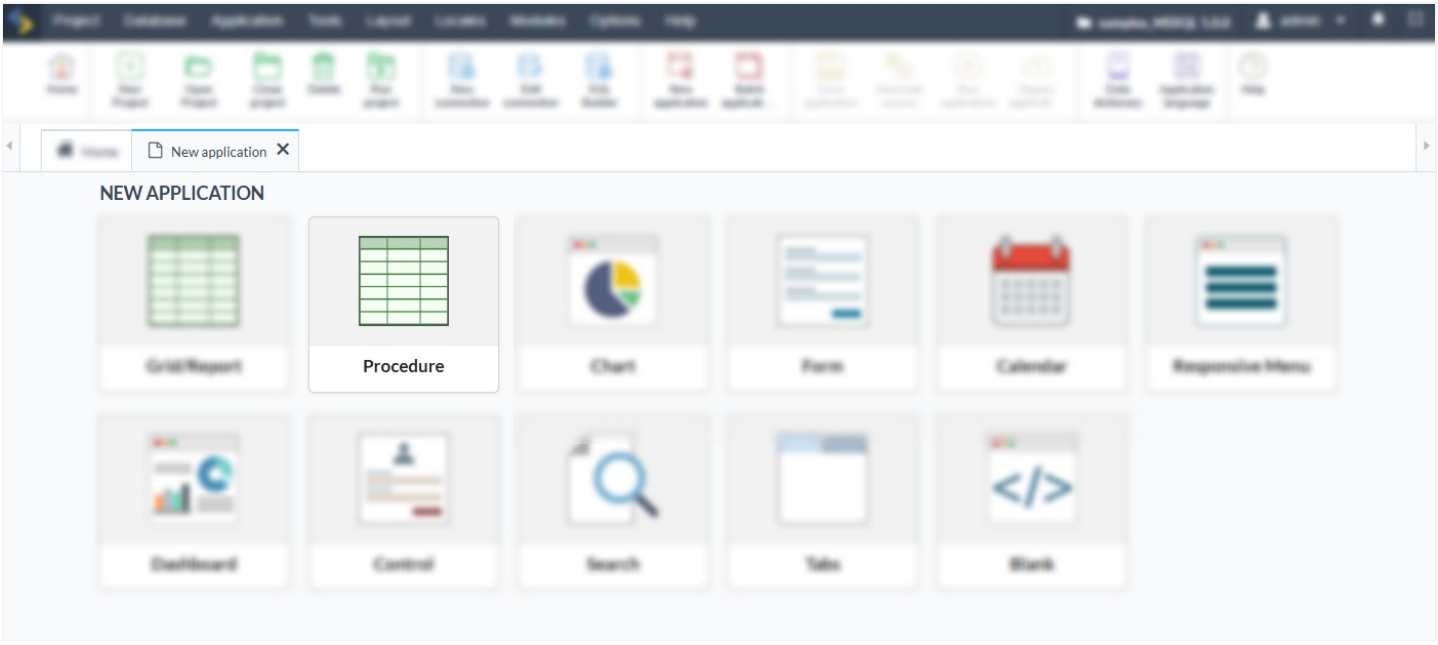
Creating a Procedure Application

New Application

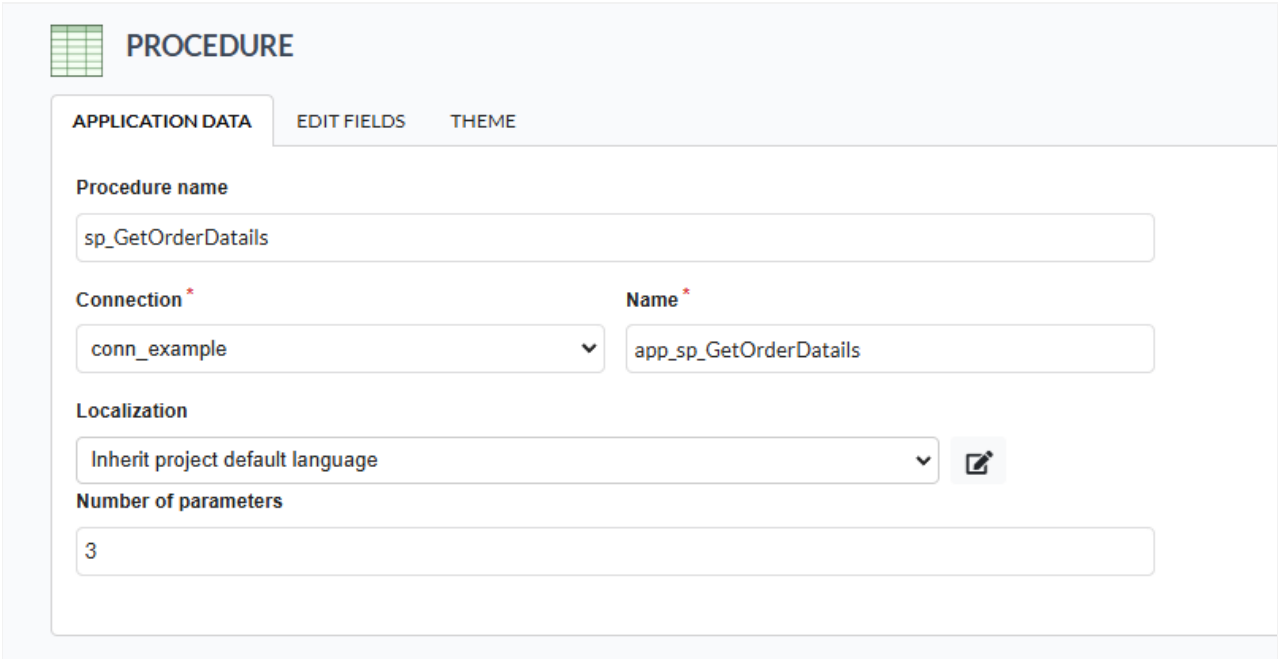
The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.

This application is available only for MSSQL Server, Oracle, and Db2 databases. It is compatible only when a connection to these types of databases is selected.



Application Data



Procedure Name

In this attribute, the developer must specify the name of the stored procedure created in the database.

Connection

Defines the connection used for the creation of the application. The existing connections in the project will be listed.

Name

The name of the application being created. It cannot contain special characters.

Localization

The language of the application to be created. The project's default language is automatically selected.

Number of Parameters

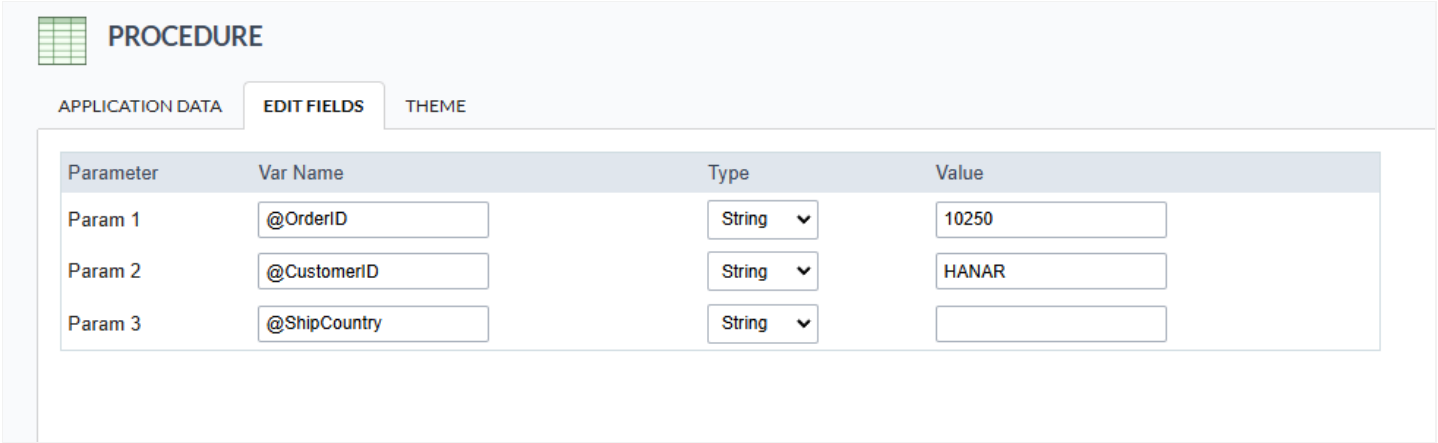
In this attribute, the developer must indicate the number of input parameters to be used. It is important to note that all parameters specified at this stage become mandatory for the application's functionality.

After creation, it is possible to modify the parameters (add or remove) by accessing the SQL menu.

If the stored procedure has no input parameters, the developer must enter 0 in the **Number of Parameters** field.

Edit Fields

In this tab, we will configure the parameters specified during the application's creation so they can be correctly defined.



Parameter

Displays the number of parameters specified in the **Application Data** section, under the **Number of Parameters** attribute.

Var Name

In this attribute, the developer must provide the name of the variables to be used in the application. These names should correspond to the parameter names defined in the database's stored procedure.

Type

Select the data type corresponding to the stored procedure's parameter: **INT** or **VARCHAR**.

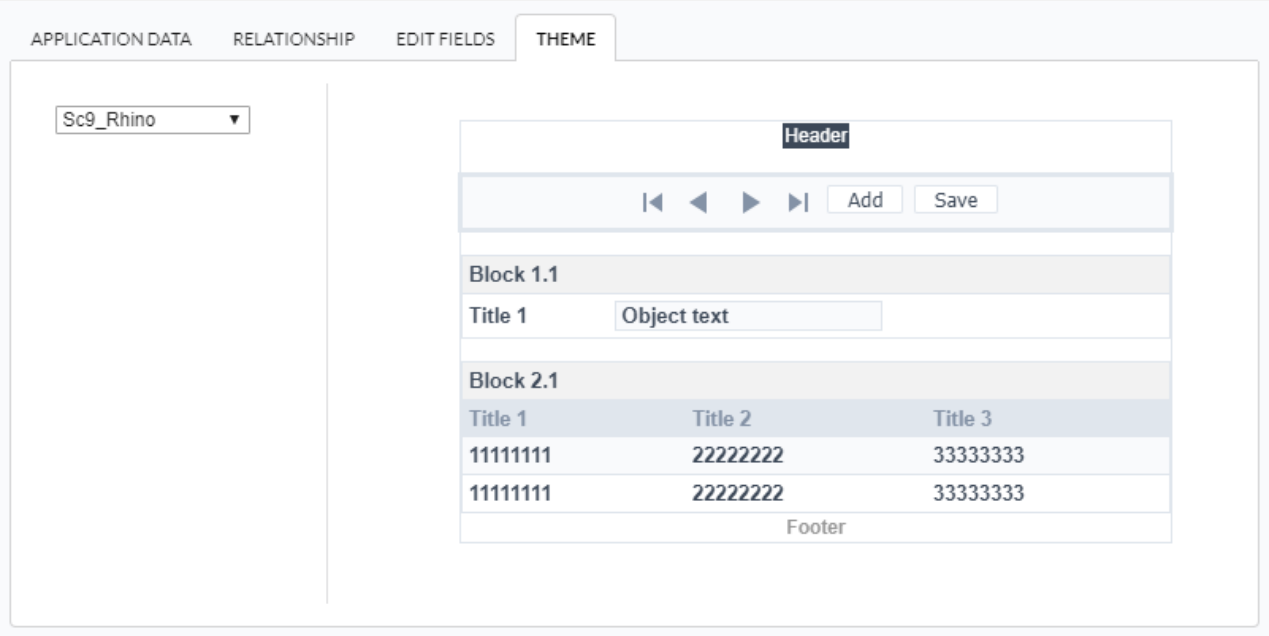
Value

Specify a value to validate the stored procedure provided.

Theme

On this screen, you can select the theme that the application will use. The default project theme, which can be checked and

modified in **Project** > **Properties**, is automatically selected.



Related Videos ▷

Email Export

It allows you to send the generated export file by email. For that, you may be using SMTP or integration with **Mandrill** and **SES**.

For more information about Mandrill click here

For more information about Amazon SES click here

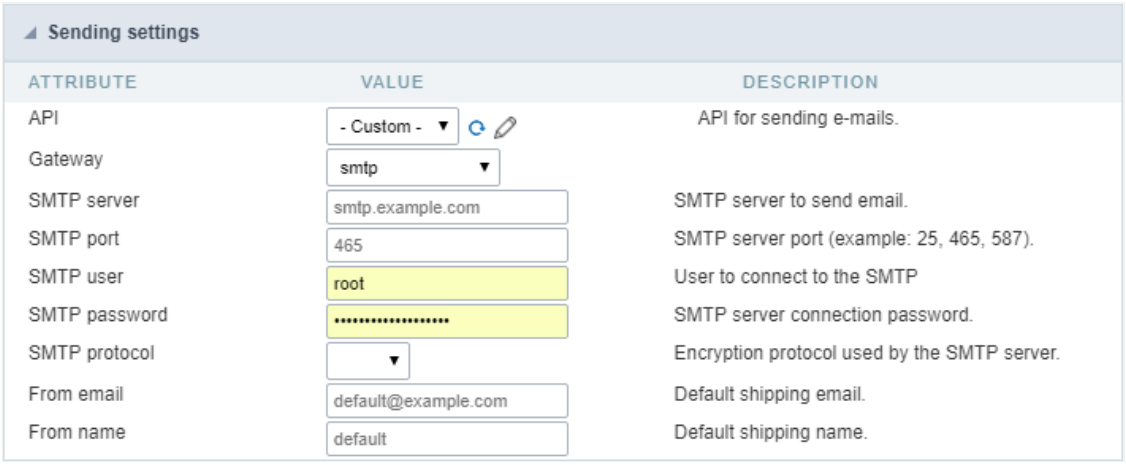
Sending Options

We must configure how to send the email or select an API already set in Tools> API. Click Here and see how to configure.

SMTP

SMTP is the standard protocol for sending emails over the Internet, and each provider has its SMTP.

For more information about SMTP settings click here



- **API**: Allows you to select an API that is already configured in **Tools> API** or set a new one here by selecting **- custom -**. In this case you will see some options according to the selected Gateway.

SMTP Server

SMTP server address for the used provider.

Port SMTP

Define the Port of the mail server. Use the port 465 for security with SSL, 587 for security with TLS, or port 25 for port without security. By default, Scriptcase uses port 25.

User SMTP

SMTP server user.

Enter SMTP

SMTP server user password.

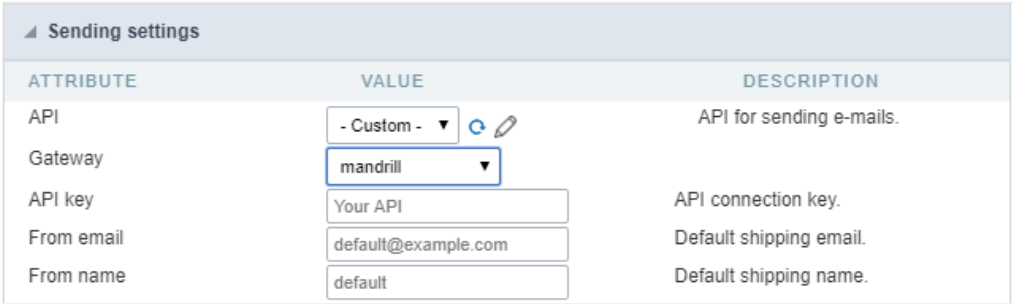
Protocol SMTP

Select the security protocol. If no value is declared, Scriptcase uses the No Security protocol as default.

- **E-mail**: Sets the origin email, which will send the emails.
- **Name**: Name that will be displayed in the sent email.

Mandrill

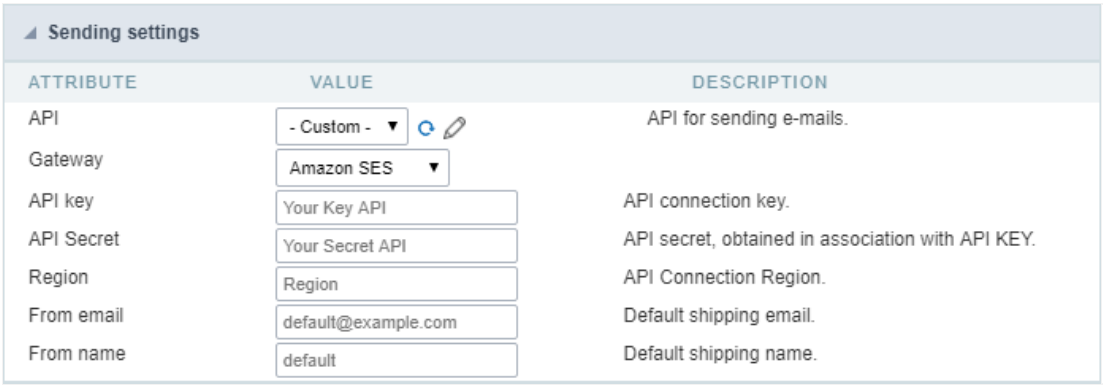
Mandrill is a transactional email API for MailChimp users, ideal for sending data-driven emails.



- **API**: Allows you to select an API that is already configured in **Tools> API** or set a new one here by selecting **- custom -**. In this case you will see some options according to the selected Gateway.
- **API KEY**: Enter the key you obtained when setting up your API.
- **E-mail**: Sets the origin email, which will send the emails.
- **Name**: Name that will be displayed in the sent email.

Amazon SES

Amazon Simple Email Service (Amazon SES) is an email sending service designed to assist in sending marketing emails, notifications, and transactional messages.



- **API**: Allows you to select an API that is already configured in **Tools> API** or set a new one here by selecting **- custom -**. In this case you will see some options according to the selected Gateway.
- **API KEY**: Enter the key you obtained when setting up your API.

API Secret

Enter the secret access key of your Amazon SES account.

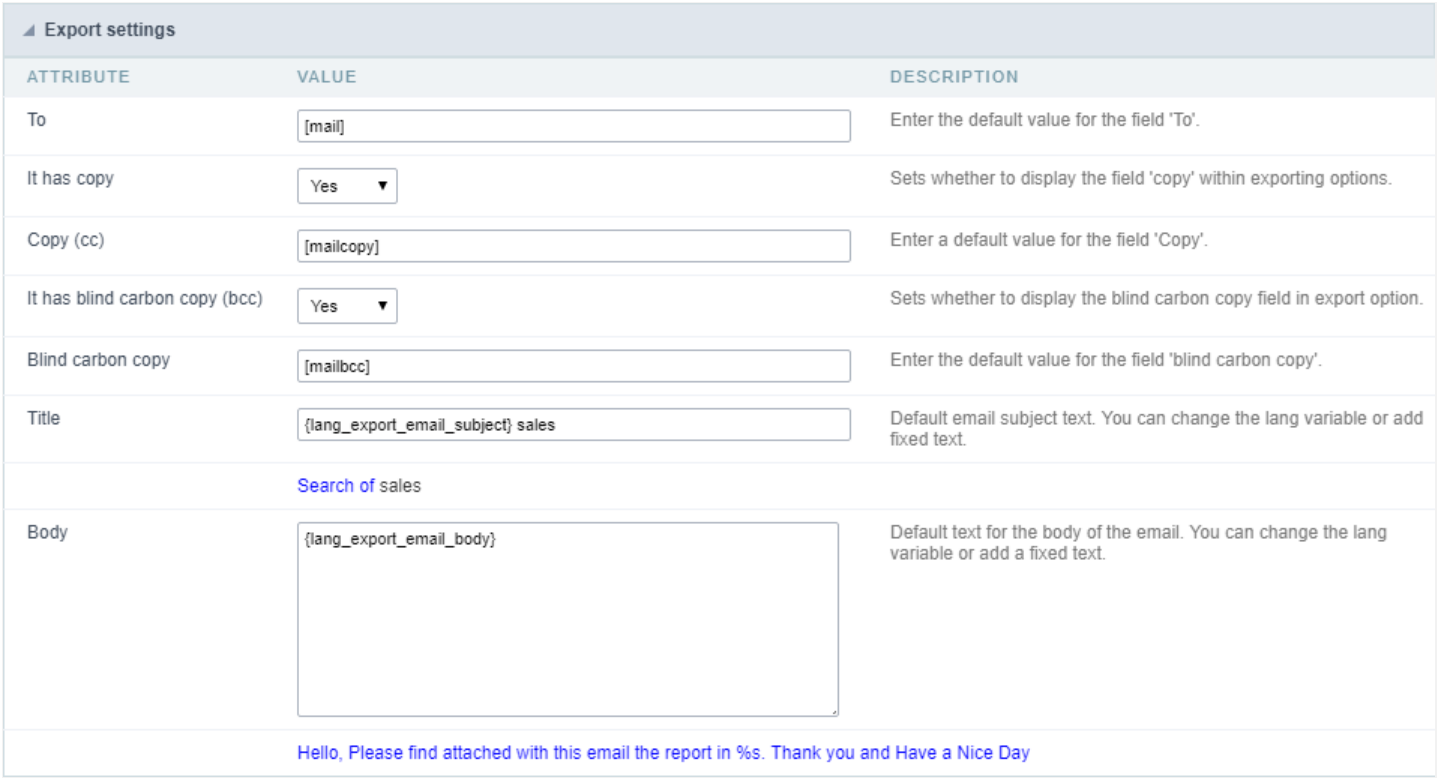
Region

Amazon SES has endpoints in several regions, to reduce network latency, inform the region of the endpoint closest to your application. See the regions.

- **E-mail**: Sets the origin email, which will send the emails.
- **Name**: Name that will be displayed in the sent email.

Export Settings

In these settings, we define the settings of the email sending interface when exporting, and we can define a default email subject and body.



To

Enter the default destination email.

Use Copy (CC)

Defines whether or not to display the 'copy' field within the export options.

- **No**: There will be no possibility to send email with copies.
- **Yes**: The field will be displayed within the export options and we can define a default email for copying. The end user can view this email.
- **Hidden**: The field will not be displayed within export options and we can set a default email for copying.

Copy (CC)

Enter the default email for the copy field. This option can also be empty, or you can use a global variable containing an email list.

Using Hidden Copy (BCC)

Sets whether or not to display the 'hidden copy' field within the export options.

- **No**: There will be no possibility to send email with copies.
- **Yes**: The field will be displayed within the export options and we can define a default email for copying. The end user can view this email.
- **Hidden**: The field will not be displayed within export options and we can set a default email for copying.

Copy (BCC)

Enter the default email for the hidden copy field. This option can also be empty, or you can use a global variable containing an

email list.

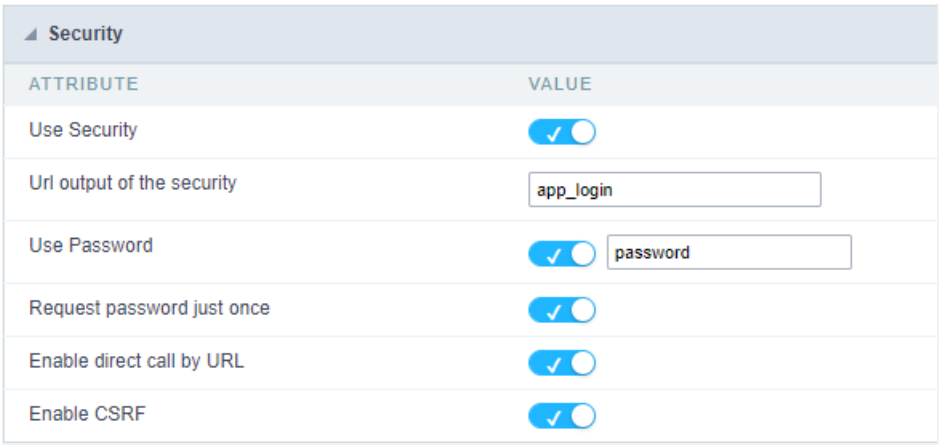
Subject

Default email subject text. You can change the lang variable or add static text.

Color

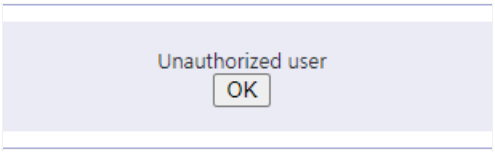
Default body text for the email. You can change the lang variable or add static text.

Security



Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application.



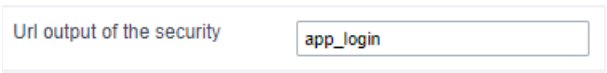
When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

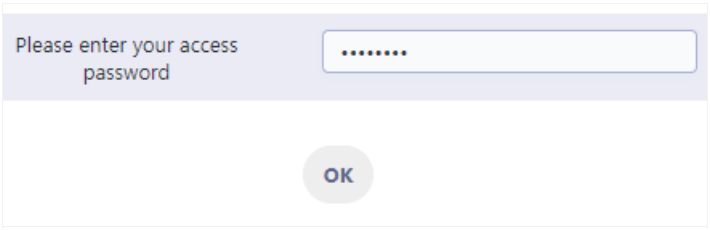
Defines which application the user will be redirected to after the “unauthorized user” warning.



Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.



The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

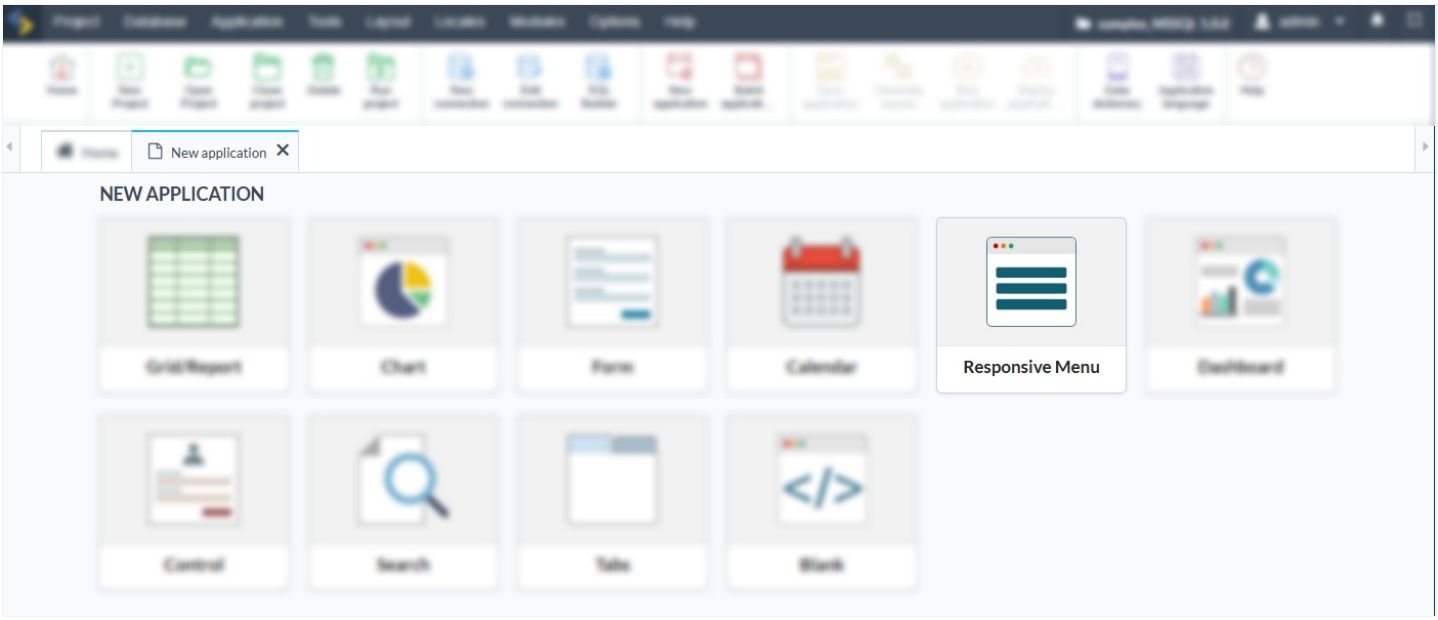
When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Creating a Responsive Menu

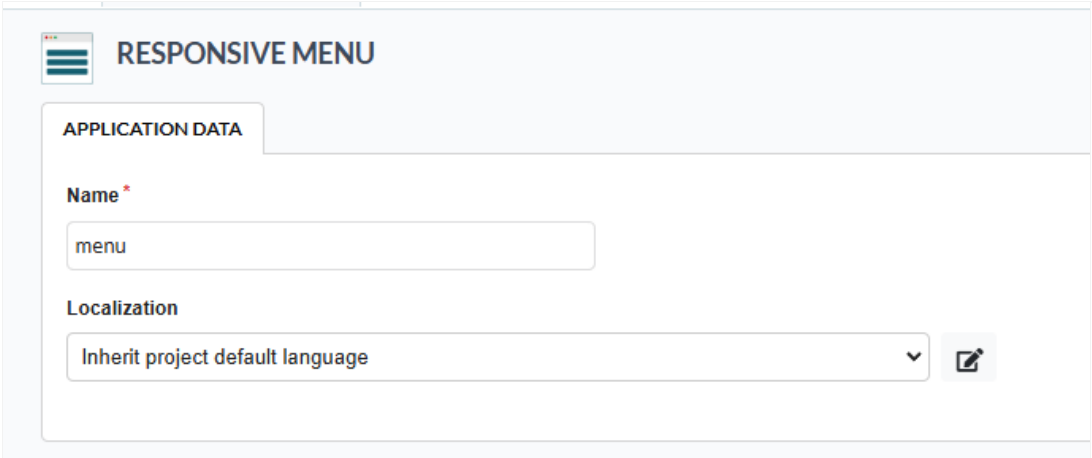
New Application

The process of creating an application is quite simple and intuitive.

You only need to select the type of application you want to generate from the list of applications available in Scriptcase, and it will be generated automatically.



Application Data



Name

The name of the application being created. It cannot contain special characters.

Localization

The language of the application to be created. The project's default language is automatically selected.

Responsive Menu Settings

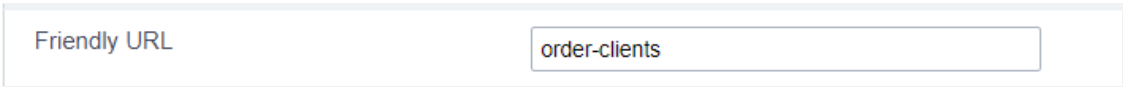
Menu Settings

The menu settings allow to define the attributes of the application display, according to the following:

Menu configuration Interface.

Friendly URL

This attribute defines the application's friendly URL.



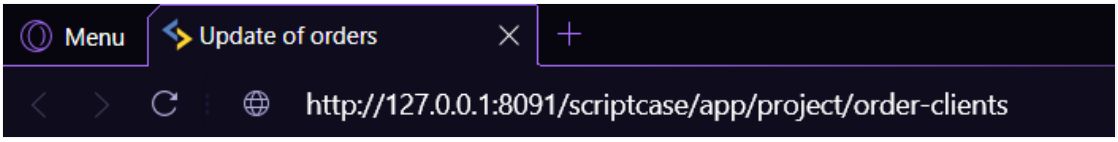
Alphanumeric characters and some special characters are allowed, such as: **hyphen** (-), **underscore** (_), **comma** (,), and **dot** (.).

The use of accents or spaces is not allowed.

Some recommendations for definition are:

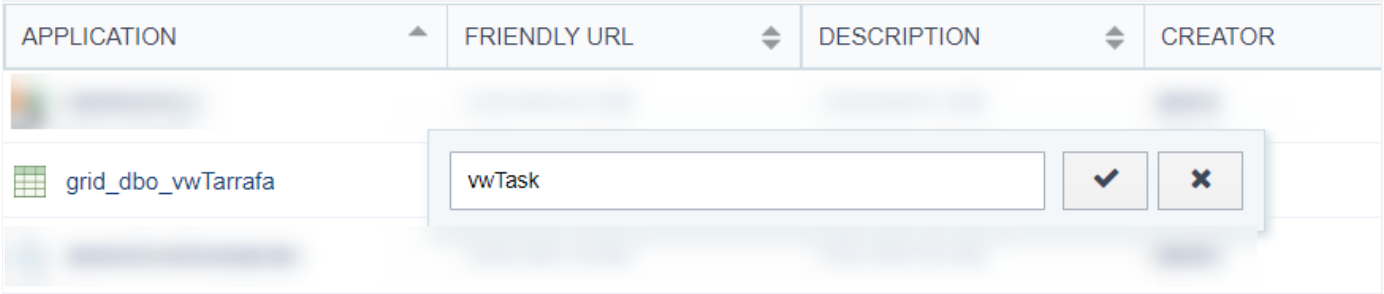
- Use keywords for identification.
- Use hyphens to separate words.
- Use only lowercase letters.
- Avoid using dates.

Example of an application using a Friendly URL



The friendly URL can also be defined in the application list on the project's initial screen.

See the example below



Hide menu items

This attribute defines the behavior of menu items based on the configuration set in the URL.

When the attribute is enabled

Menu items will be hidden in the following situations:

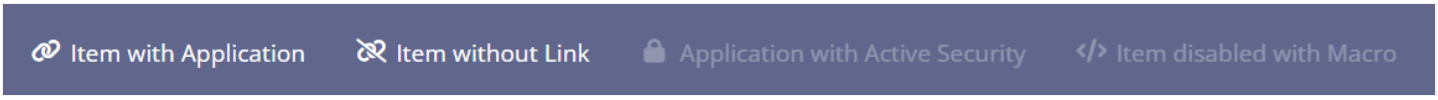
- **No link configured**: When the item does not have an associated link.
- **Link to an application with active security**: When the link points to an application with active security and the user does not have access permission.
- **Disabled by macro**: When the item is disabled using the `sc_menu_disable` macro.

When the attribute is disabled

Menu items will always be displayed, but with different behaviors depending on the situation:

- **No link configured**: It will be displayed as a regular item without visual changes.
- **Link to an application with active security or Disabled by macro**: It will be displayed but in a disabled state.

Example of items with the attribute disabled

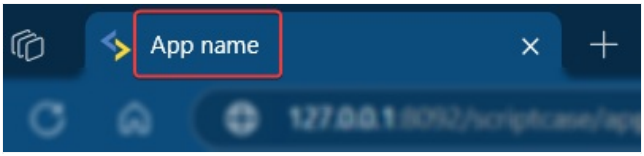


Application Title

Defines the application title, which can be set dynamically using langs or with a fixed text.

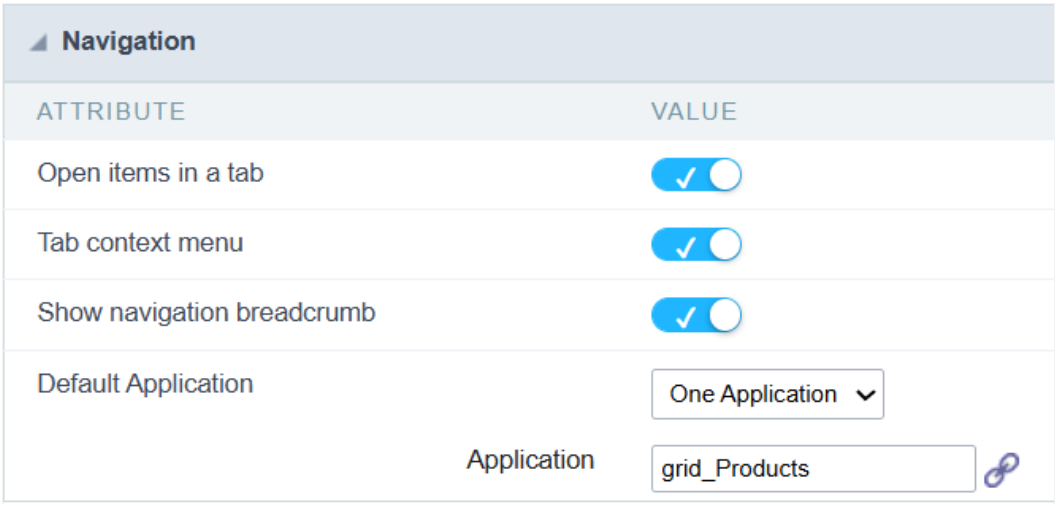
In this option it is also possible to use langs, for a system with multiple languages. Click here and see how to configure the langs

Example of the application title



Navigation

Defines some display options for the menu, such as using navigation paths (breadcrumb), tabs, and the initial application.

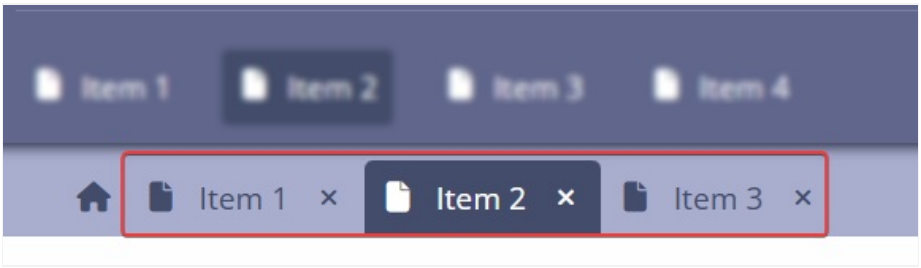


Menu Navigation Configuration Interface.

Open Items in Tabs

Defines the behavior of the menu when opening items.

When the attribute is enabled items will be opened in tabs, as shown in the image below.

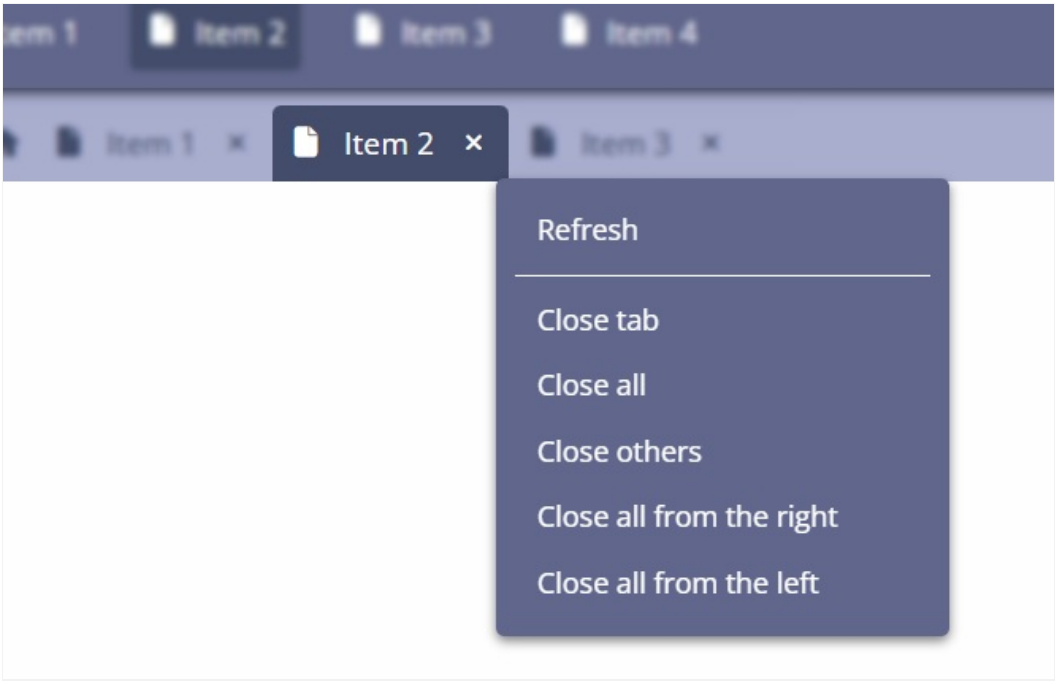


Application with tab navigation.

Tab Context Menu

Enables the display of a context menu when right-clicking on one of the open tabs in the menu application, making it easier for the user to interact with open tabs.

Example of the context menu



The displayed options are:

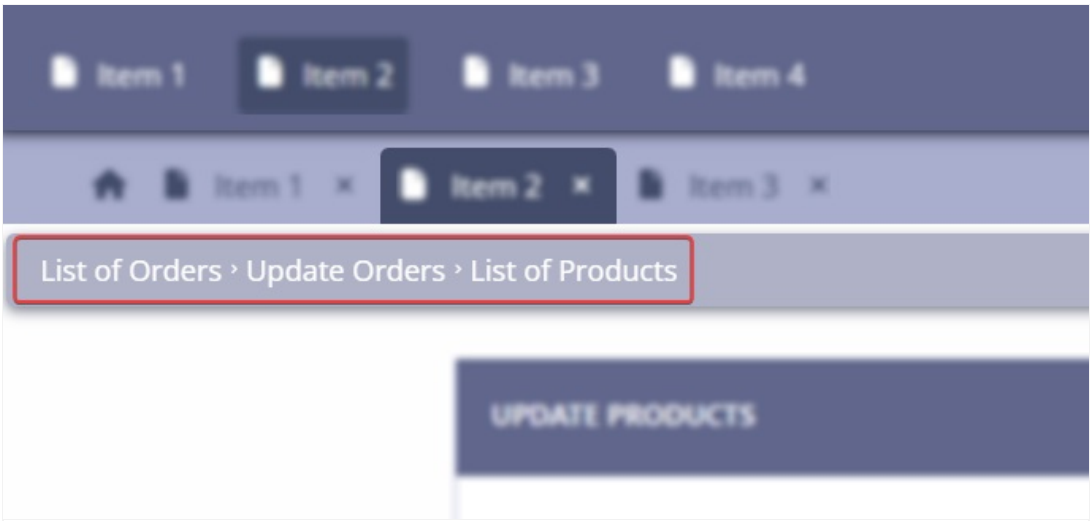
- **Reload** - Reloads the tab where the context menu was opened.
- **Close Tab** - Closes only the tab where the menu was opened.
- **Close all** - Closes all open tabs.
- **Close others** - Closes all other tabs, keeping only the tab where the menu was opened.
- **Close all to the right** - Closes all tabs positioned to the right of the tab where the menu was opened.
- **Close all to the left** - Closes all tabs positioned to the left of the tab where the menu was opened.

This attribute is displayed only when the menu is configured to open in tabs.

Show Navigation Path

This attribute enables the display of the navigation path (breadcrumb), which will be shown when navigating between links in an application.

Example of the breadcrumb



Default Application

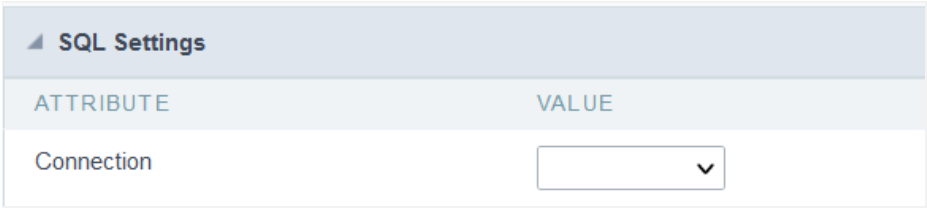
Defines whether the menu will have an application or a default URL, always displayed when the menu is opened. This attribute can be set as:

- **Blank** - The menu will start with a blank screen.
- **An application** - The menu will open with an application already loaded.
- **URL (Web Address)** - The menu will display a web page as the initial screen.

If the option to use tabs is enabled, the initial application will be displayed in the home tab, which cannot be closed.

Responsive Menu SQL Settings

SQL Settings



The Connection field defines the connection that will be used by the application. In Menu type application, it is not mandatory to select a connection.

By default, the connection field has no value after creating a menu type application.

In menu applications, it is only necessary to select a connection to use some code in the events, for example, to create a dynamic menu.

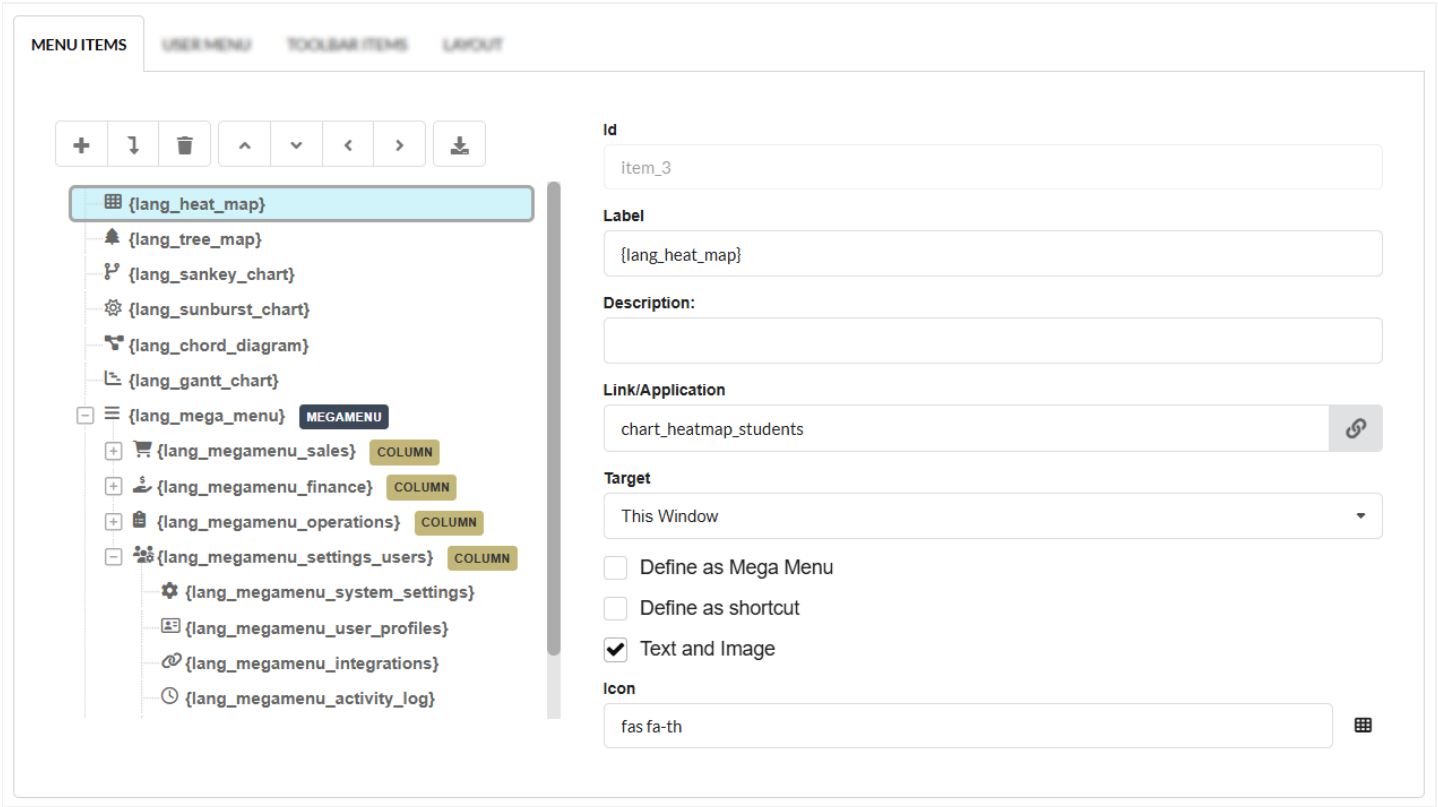
Menu Items

On this screen, the developer manages the entire structure of the menu application for the system. It is possible to define navigation with menu items and sub-items, choose whether to display the user menu, configure the components that will be shown in the toolbar, and adjust all layout settings.

Menu items

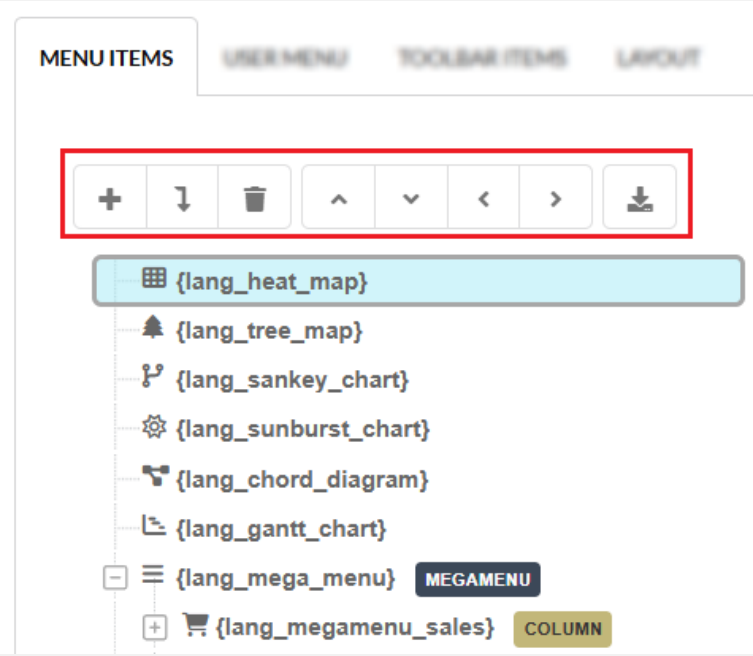
In the **Menu Items** tab, you can define the items and sub-items that make up the menu navigation structure, as well as their visual aspects, such as the **Label** and **Icon**.

On mobile devices, if a menu item has both a link and sub-items, the link will be ignored to display the sub-items.



Menu Structure

On the left column, the developer can organize the menu structure using the management buttons.



Add Item

Allows adding an application to the menu.

Add Sub-Item

Allows adding a sub-item to the menu.

Remove

Deletes a menu item. Select the item or sub-item and click the remove button.

Move Up

Moves a menu item or sub-item up in the menu structure.

Move Down

Moves a menu item or sub-item down in the menu structure.

Move Left

Moves a menu item or sub-item to the left, reducing its hierarchy level.

Move Right

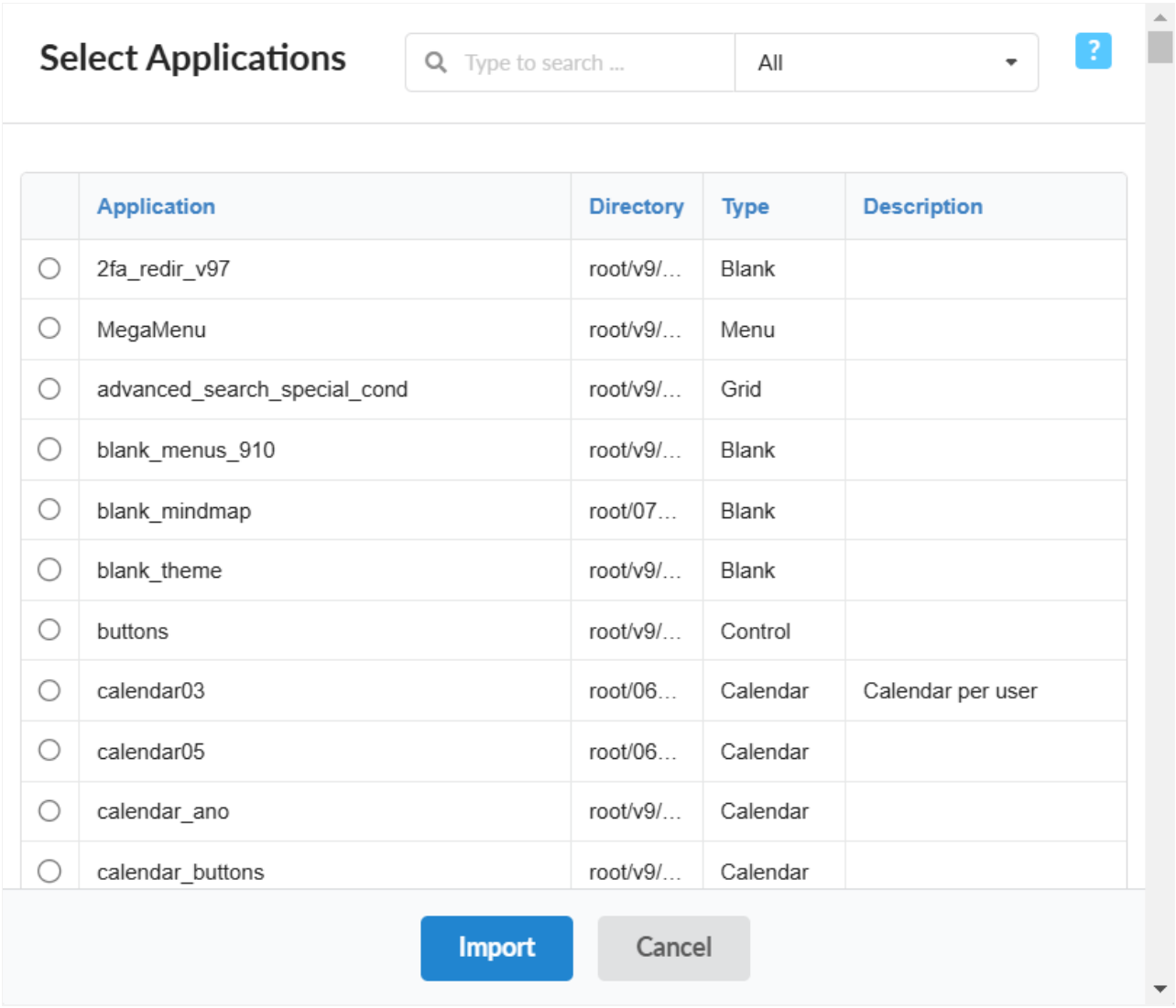
Moves a menu item or sub-item to the right, increasing its hierarchy level.

Import Applications

Allows creating a link to an application. The user can click the search button to select an application.

Application import allows the user to select applications from a list to be added. After selection, the menu levels can be organized using the available structuring buttons.

Application List for Import

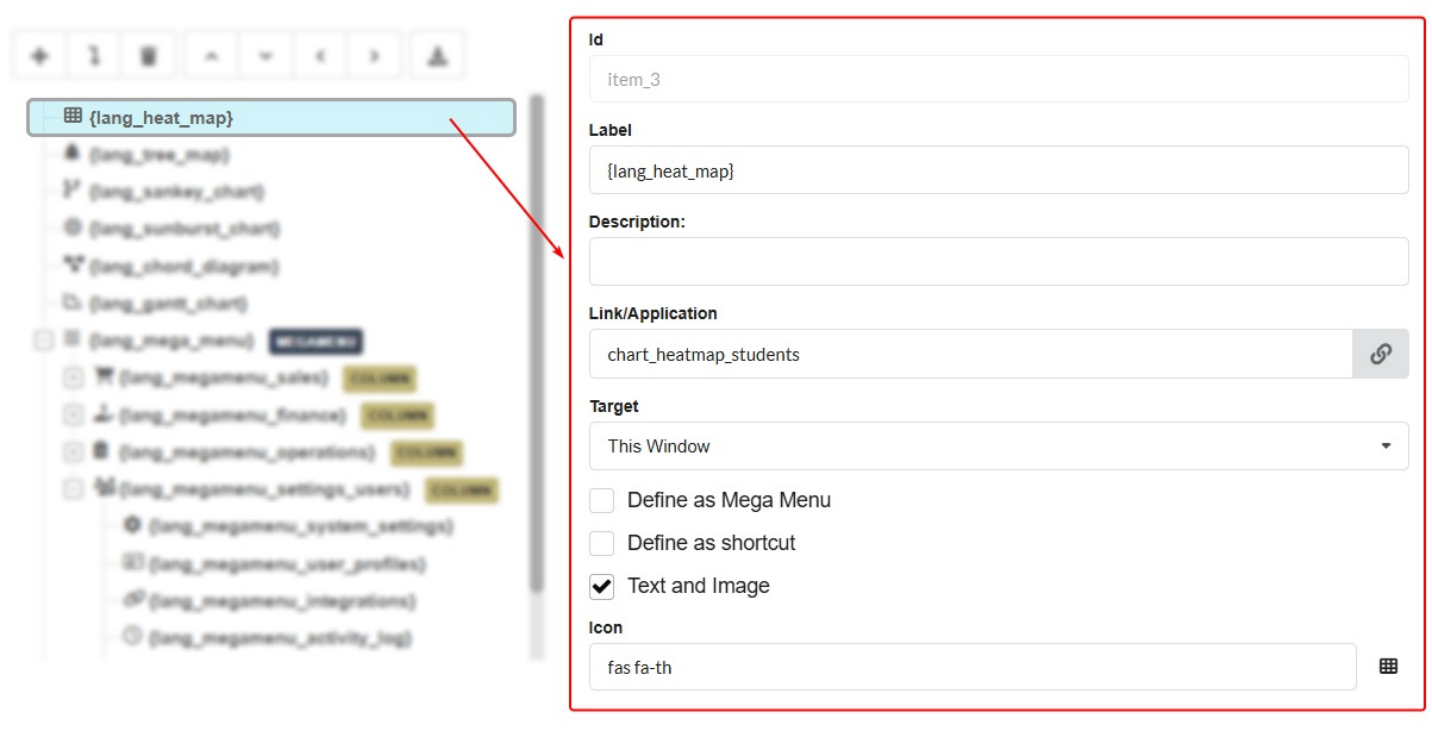


Application

Import Interface.

Selected Item Configuration

On the right column, you can configure the properties of the selected item.



ID

Displays the item creation ID. This information is useful for managing items using compatible macros.

Label

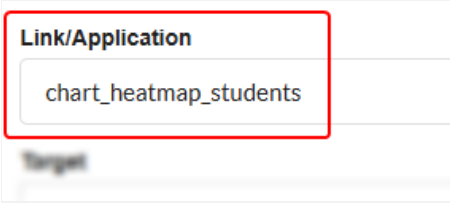
Defines the title of the item that will be displayed to the end user. The **Label** can be set as a fixed text or use a **lang** for internationalization.

Description

Allows adding a description to the menu item, making it easier to identify.

Link/Application

Defines the application or external link associated with the **Menu** item. The developer can use the search button to select an application from the project.



Application Import Interface.

Target

Defines how the application will open in the menu, either in the same window or a new one.

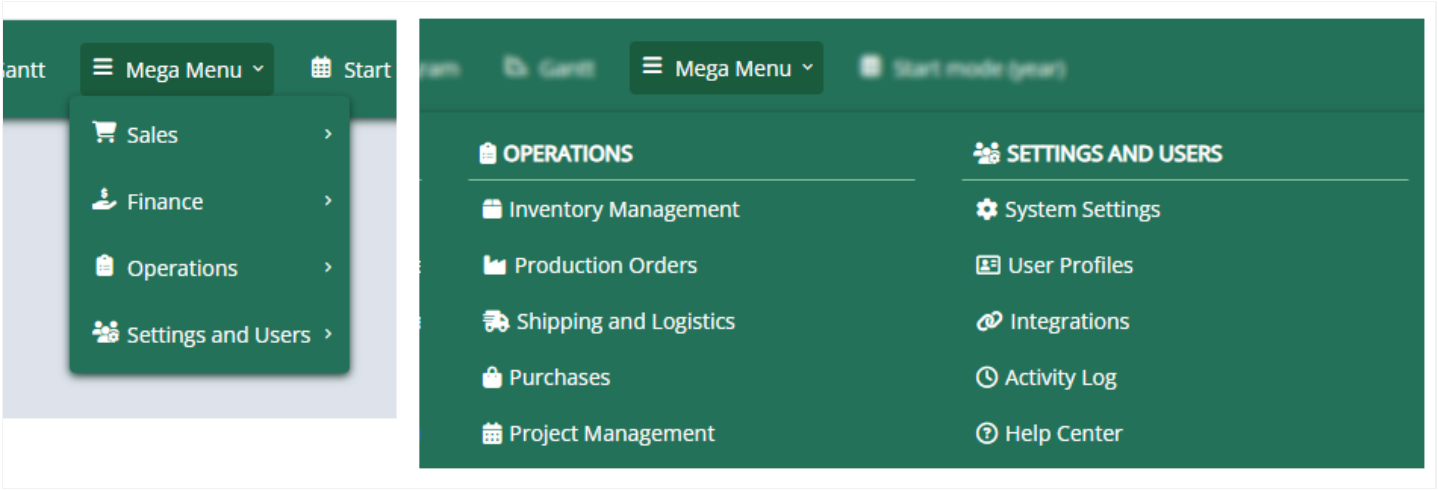
Define as Mega Menu

Enables the item as a **Mega Menu**, automatically organizing level 1 subfolders into columns.

Important: This option is only available for root-level items.

Display Comparison Mega Menu

On the left, the standard menu item; on the right, the same item configured as a Mega Menu.



Define as shortcut

Sets the icon as a shortcut.

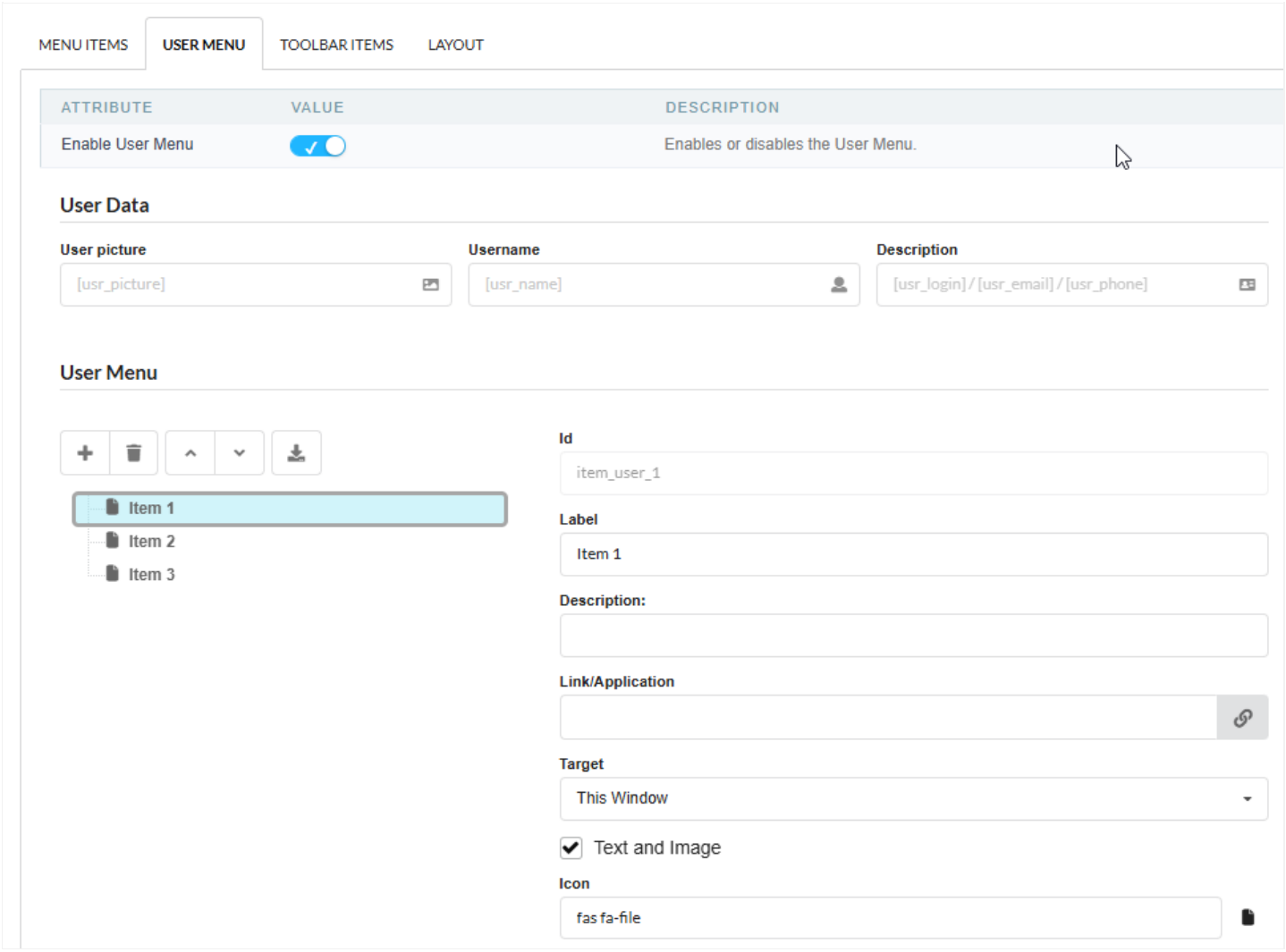
Text and Image

Displays the text along with the image.

Icon

The image that will be displayed as the menu item icon. The user can click the search button to select the desired image. This field is optional.

User Menu



These settings concern the items that will be displayed when clicking on the user icon in the menu.

Enable User Menu

Enables or disables the user menu

Use picture

Picture that will be displayed.

Username

User name that will be displayed.

Description

Description for the menu user.

Insert item

Used for the user to include the application in the menu.

Remove

Removes an item from the menu. Select the item or sub-item, and then click the remove button.

Move up

Allows you to move a menu item or sub-item up.

Move down

Allows you to move a menu item or sub-item down.

Import Applications

Allows you to create a link with an application. User can click on search button to select application.

Label

Title of the item that will be displayed.

Link

Name of the application that will create the link. User can click on search button and select application.

Target

Defines how the application will be displayed in the menu, which can be: the same window or another window.

Text and image

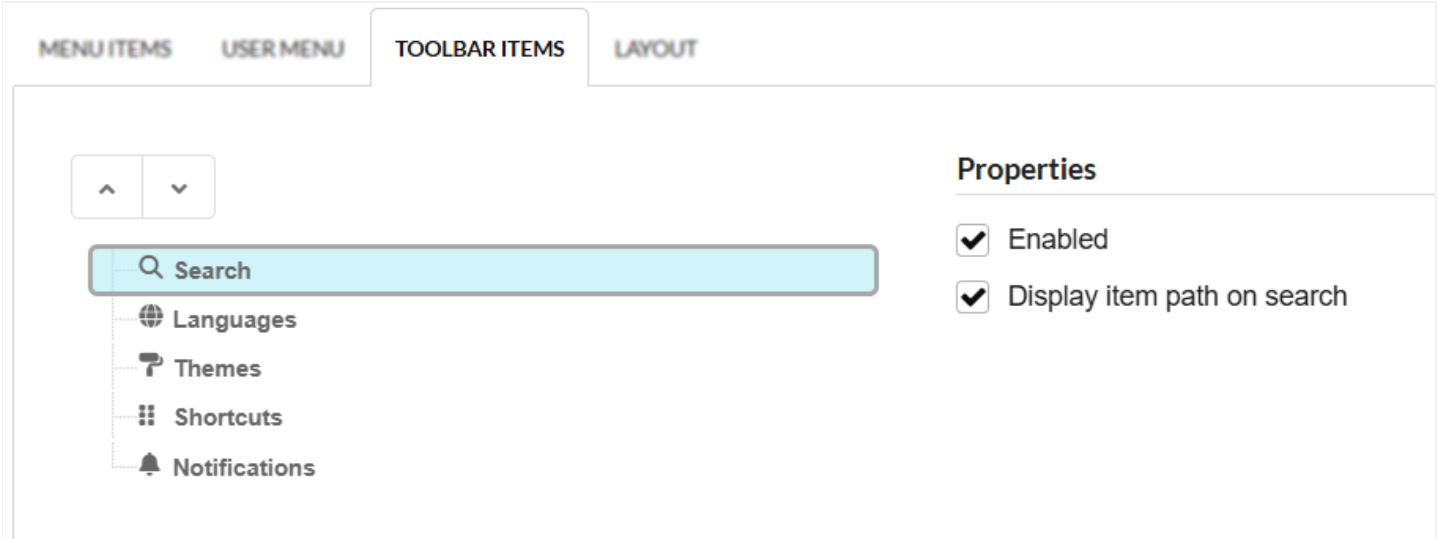
Displays the text together with the image

Icon

Name of the image that will be displayed as an icon in the menu item. User can click on search button to select desired image. Optional field.

Toolbar

It enables the standard menu items to be displayed.



Sorting the items

Move up

Allows moving a menu item up.

Move down

Allows moving a menu item down.

Properties

Search

- **Enabled** - Allows enabling or disabling the search item in the menu's toolbar.
- **Display item path in search** - Defines whether the path to the item will be displayed alongside the item's name after the search.

Languages

- **Enabled** - Allows enabling or disabling the languages item in the menu's toolbar.

Themes

- **Enabled** - Allows enabling or disabling the themes item in the menu's toolbar.
- **Available themes** - List of selected themes to be displayed in the application.
- **All** - Selects all themes.
- **None** - Removes all themes, eliminating the button from the running application.

Shortcuts

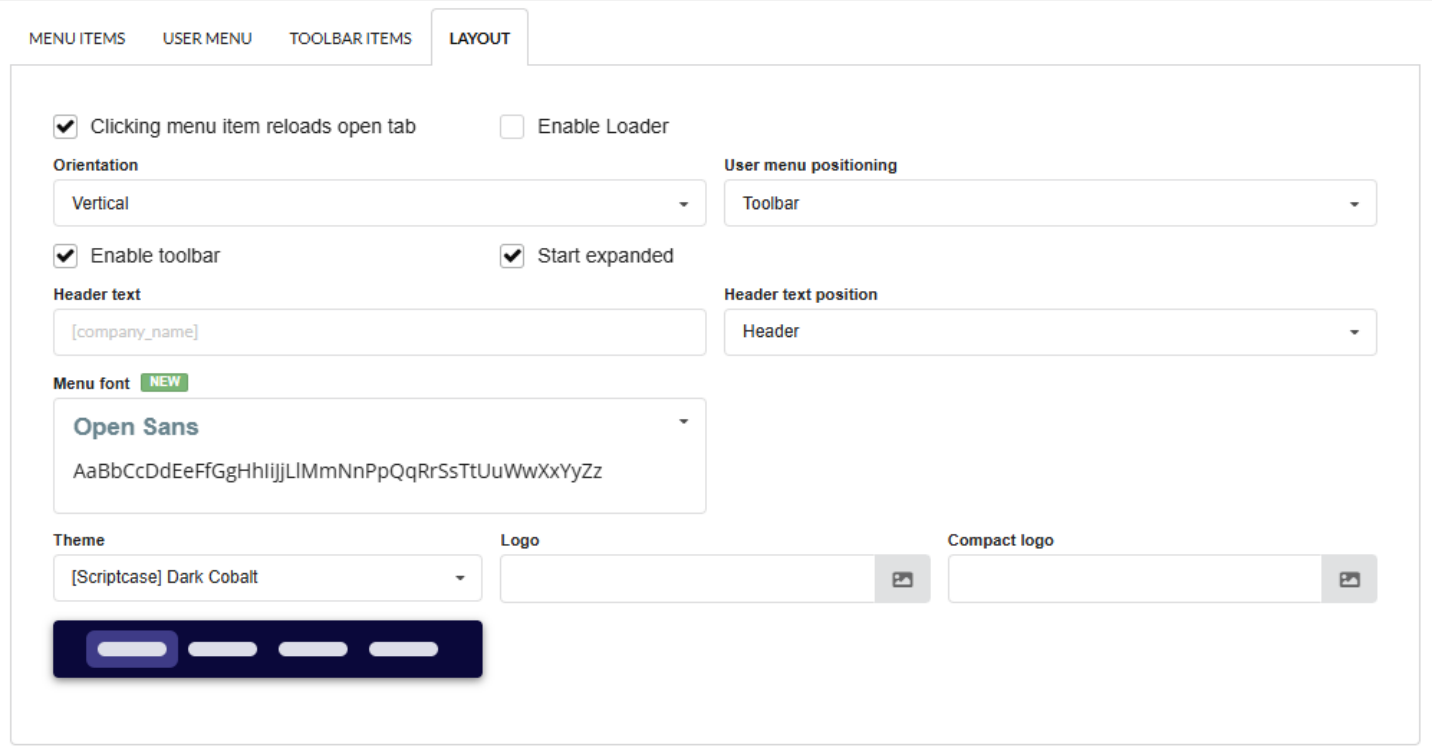
- **Enabled** - Allows enabling or disabling the shortcuts item in the menu's toolbar.
- **Show Shortcuts title** - Configures whether the title of the shortcuts list will be displayed in the running application.
- **Shortcuts order** - Allows sorting the shortcut items.

Notifications

- **Enabled** - Allows enabling or disabling the notifications item in the menu's toolbar.
- **Notification listing application** - Link to the application that will display the notification list in the menu item.
- **Notification refresh interval** - Time in milliseconds for the notification list to be updated.
- **Row limit** - Maximum number of notifications displayed on the icon.
- **Connection** - Defines the connection where the notification system tables are located.
- **User ID variable** - Name of the global variable that stores the user's login.
- **Table mapping** - For the notification module to function in the menu, it is necessary to link the **Notification**, **inbox**, and **users** tables for proper display of notifications on the icon.

When creating a notification module and selecting an already created menu, the fields should be automatically filled.

Layout



Layout

Defines how the menu will be displayed.

Clicking menu item reloads open tab

Specifies whether the open tab will be reloaded when the menu item is clicked.

Enable Loader

Enables the loading animation when a menu item is being loaded.

Orientation

Determines if the menu will be displayed horizontally or vertically.

User menu positioning

Specifies the position of the user menu when the main menu is set to vertical.

Enable toolbar

Enables the toolbar in the menu.

Start expanded

Specifies whether the menu will start in an expanded or collapsed state. Available only for vertical menus.

Header text

Sets the text to be displayed in the menu header.

Header text position

Specifies the position of the menu header text.

Menu font

Defines the font used in the menu.

Theme

Menu theme to be defined.

Logo

Logo that will be displayed.

Compact Logo

Compressed image that will be displayed as the logo in the menu

Security

Security

Use security

With this option enabled, when accessing the application, an “unauthorized user” message will be displayed, preventing access to the application.

When enabled, application access is only possible through the macro **sc_apl_status** or through **security module**.

Click here to access the macro documentation sc_apl_status.

> In the development environment, the use of the above options for testing purposes may be suppressed, preventing the password being requested or access being blocked at each execution of the applications. To do this, disable the options “Enable security usage” and “Enable password usage” in the menu **Configuration > My Scriptcase**.

Security exit url

Defines which application the user will be redirected to after the “unauthorized user” warning.

Use password

When enabling the option, you will be asked to define a password for accessing the application.

When running the application using the active password, you will be asked for the password before accessing the application.

> The password will be requested regardless of the “use security” setting or the use of the security module.

Request password only once

This option defines the behavior of the use password option.

When this option is enabled, the password will be requested only once per session.

When disabling, the password will be requested every time the application will be accessed.

Allow direct calling by URL

Allows an application to be called by typing the URL directly into browsers.

When disabled, when accessing the application by typing the URL directly, the invalid data error will be displayed. With this configuration, the application can only be accessed through a menu application, for example.

Cookies

Cookies

HttpOnly

These are cookies that can be set by the server or by javascript but cannot be accessed in any way by javascript.

With the HTTPOnly parameter enabled, it is very difficult to steal a session through XSS because the javascript cannot read the content of these cookies.

A widely exploited attack in XSS is the theft of user session cookies. Theft of cookies is carried out through a javascript command that sends cookies to an external URL where the attacking user has control.

> This option is enabled by default in all applications.

Session ID

Enables the session id that will be stored in cookies on the client side. This option embeds the session id directly in the URLs

> This option is enabled by default in all applications.

Cookie Secure

With this option enabled, cookies are protected and can only be transmitted via secure communication. Therefore, they cannot be accessed through Javascript.

> The HttpOnly option must also be enabled.

Headers

Headers

Disable XSS Auditor

XSS is short for Cross-Site Scripting, which is an attack that consists of sending HTML with malicious JavaScript to a website.

It is a security attack that consists of sending HTML with malicious JavaScript to a website. If the website displays this HTML with JavaScript without filtering the malicious JavaScript code, the cookies sent by the website can be stolen and sent to an attacking website, so that it can be used to forge user sessions and access the website's user account without permissions.

Strict-Transport-Security

HSTS forces browsers to use HTTPS in the domain where it is enabled, instead of using HTTP.

When enabled, the default value is "max-age = 31536000".

The HSTS policy is set for one year (31536000 seconds). This period specifies the time the browser will access the server via HTTP

X-Frame-Options

The HTTP X-Frame-Options response header can be used to indicate whether or not the browser should render the page in a <frame>, `<iframe>` `,` < embed> '' or ''. This header provides protection against clickjacking, known as 'click swipe' - this vulnerability can steal relevant information and data from the user.

X-Frame-Options does not allow the rendering of a page in a frame, ensuring that the content of your page is not embedded in other sites

- **SAMEORIGIN**: The SAMEORIGIN directive allows the page to be loaded in a frame from the same source as the page itself. This option must be used so that the application with clickjacking protection can be used in the menu application.
- **DENY**: This directive completely disables page loading in a frame.

  If no option is selected, the application can be used on any `<frame>` , <iframe> `,` or `<object>, allowing the application to be incorporated into another website.

X-Content-Type-Options

X-Content-Type-Options is used to protect against MIME detection vulnerabilities. With this option enabled, it prevents browsers from interpreting the page content (sniffing) and executing the data as code / tag. These vulnerabilities can occur when uploading a text file with a javascript code and the browser reading the content that is in the file and executing, even though it is just text and not part of the code.

With this enabled, the HTTP X-Content-Type-Options header is defined, by default, as "no detection".

Referrer-Policy

no-referrer.

The Referer header will be omitted completely. No reference information will be sent with requests.

no-referrer-when-downgrade.

This is the default behavior when no policy is specified, or if the value provided is invalid.

origin.

Only send the origin of the document as a reference. For example, a document at https://example.com/page.html will send the reference https://example.com/.

origin-when-cross-origin.

Sends the origin, path and query string when performing a same-origin request, but only sends the origin of the document in other cases.

same-origin.

The reference will be sent to sources on the same site, but requests between sources will not send reference information.

strict-origin.

Only send the document source as a reference when the security protocol level remains the same (HTTPS → HTTPS), but do not send it to a less secure recipient (HTTPS → HTTP).

strict-origin-when-cross-origin.

Sends the source, path and query string when performing a request from the same source, only sends the source when the security protocol level remains the same during an inter-source request (HTTPS → HTTPS), and sends no header to recipients less secure (HTTPS → HTTP)

unsafe-url.

Sends the source, path and query chain when performing any request, regardless of security.

Feature-Policy

The value of this header is a policy or set of policies that you want the browser to respect for a particular source.

The source whitelist can take on several different values:

- `*:` The feature is allowed in top-level navigation contexts and in nested navigation contexts (iframes).
- `'self':` The feature is allowed in top-level navigation contexts and nested navigation contexts from the same source. It is not allowed in documents of cross origin in nested browsing contexts.
- `'none':` The feature is not allowed in top-level browsing contexts and is not allowed in nested browsing contexts.
- `<origin (s)>:` specific sources for which to enable the policy (for example https://example.com).

Example 1 - Using only one directive Let’s say you want to prevent all content from using the geolocation API on your site. You can do this by submitting a restricted ‘none’ whitelist to the geolocation resource:

Feature-Policy: `geolocation 'none'`

Example 2 - Using more than one directive Functionalities within a policy are separated by semicolons.

Feature-Policy: unsized-media 'none'; geolocation 'self' https://example.com; camera *; `

Directives

accelerometer

Controls whether the current document is allowed to collect information about the acceleration of the device

via the Accelerometer interface.

ambient-light-sensor

Controls whether the current document is allowed to collect information about the amount of light in the environment around the device through the AmbientLightSensor interface.

autoplay

Controls whether the current document is allowed to play the requested media automatically via the HTMLMediaElement interface. When this policy is disabled and there is no user action, the Promise returned by HTMLMediaElement.play () will reject it with a DOMException. The autoplay attribute on <audio> '' and '' elements will be ignored.

battery

Controls whether the use of the Battery Status API is allowed. When this policy is disabled, the Promise returned by Navigator.getBattery () will reject it with a NotAllowedError DOMException.

camera

Controls whether the current document is allowed to use inputs from video devices. When this policy is disabled, the Promise returned by getUserMedia () will reject it with a NotAllowedError DOMException.

display-capture

Controls whether or not the current document is allowed to use the getDisplayMedia () method to capture screen content. When this policy is disabled, the Promise returned by getDisplayMedia () will reject it with a NotAllowedError if permission is not obtained to capture the contents of the screen.

document-domain

Control whether the current document is allowed to place document.domain. When this policy is disabled, attempts to place document.domain will fail and cause a SecurityError DOMException to be thrown.

encrypted-media

Controls whether the current document is allowed to use the Encrypted Media Extensions (EME) API. When this policy is disabled, the Promise returned by Navigator.requestMediaKeySystemAccess () will reject it with a DOMException.

execution-while-not-rendered

Controls whether tasks should be performed in frames while they are not rendered (e.g. whether a frame is hidden or display: none).

execution-while-out-of-viewport

Controls whether tasks should be performed in frames while they are outside the visible viewing window.

fullscreen

Controls whether the current document is allowed to use Element.requestFullScreen (). When this policy is disabled, the returned Promise rejects it with a TypeError DOMException.

geolocation

Controls whether the current document is allowed to use the Geolocation interface. When the policy is disabled, calls to getCurrentPosition () and watchPosition () will cause function callbacks to be invoked with a PositionError of PERMISSION_DENIED.

gyroscope

Controls whether the current document is allowed to collect information about the orientation of the device through the Gyroscope interface.

layout-animations

Controls whether the current document is allowed to show layout animations.

legacy-image-formats

Controls whether the current document is allowed to show images in legacy formats.

magnetometer

Controls whether the current document is allowed to collect information about the orientation of the device through the Magnetometer interface.

microphone

Controls whether the current document is allowed to use inputs from audio devices. When this policy is disabled, the Promise returned by MediaDevices.getUserMedia () will reject it with a NotAllowedError.

midi

Controls whether the current document is allowed to use the Web MIDI API. When this policy is disabled, the Promise returned by Navigator.requestMIDIAccess () will reject it with a DOMException.

navigation-override

Controls the availability of mechanisms that enable the author of the page to take control over the behavior of spatial navigation, or to cancel completely.

oversized-images

Controls whether the current document is allowed to download and display large images.

payment

Controls whether the current document allows the use of the Payment Request API. When this policy is disabled, the PaymentRequest () constructor will throw a SecurityError DOMException.

picture-in-picture

Controls whether the current document allows a video to be allowed to play in Picture-in-Picture mode via the corresponding API.

publickey-credentials-get

Controls whether the current document is allowed to use the Web Authentication API to retrieve already stored public key credentials, i.e. via navigator.credentials.get ({publicKey: ..., ...}).

sync-xhr

Controls whether the current document is allowed to make synchronous XMLHttpRequest requests.

usb

Controls whether the current document is allowed to use the WebUSB API.

vr

Controls whether the current document is allowed to use the WebVR API. When this policy is disabled, the Promise returned by Navigator.getVRDisplays () will reject it with a DOMException. Keep in mind that the WebVR standard is in the process of being replaced by WebXR.

wake-lock

Controls whether the current document is allowed to use the Wake Lock API to indicate that the device should not go into a power saving mode.

screen-wake-lock

Controls whether the current document is allowed to use the Screen Wake Lock API to indicate whether or not the device should darken the screen.

xr-spatial-tracking

Controls whether or not the current document is allowed to use the WebXR Device API to interact with the WebXR session.

Content-Security-Policy

Content-Security-Policy is the name of an HTTP response header that modern browsers use to increase the security of the document (or web page).

The Content-Security-Policy header allows you to restrict resources such as JavaScript, CSS or just about anything the browser loads.

Directives

default-src

The default-src directive defines the standard policy for fetching features such as JavaScript, Images, CSS, Fonts, AJAX requests, Frames, HTML5 Media. Not all directives go back to default-src. Refer to the Source List Reference for possible values.

DFAULT-SRC Policy Example default-src 'self' cdn.example.com;

script-src

Defines valid JavaScript sources.

SCRIPT-SRC Policy Example script-src 'self' js.example.com;

style-src

Defines valid fonts for style sheets or CSS.

EXAMPLE__ STYLE-SRC POLICY style-src 'self' css.example.com;

img-src

Defines valid image sources.

Example of IMG-SRC Policy img-src 'self' img.example.com;

connect-src

Applies to XMLHttpRequest (AJAX), WebSocket, fetch (), `<a ping> '' 'or EventSource. If not allowed, the browser emulates a 400 HTTP status code. __Connect-SRC Policy Example__ connect-src ‘self’; ‘’

font-src

Defines valid fonts for font resources (loaded via @ font-face).

FONT-SRC Policy Example font-src font.example.com; `

object-src

Defines valid plug-in sources, for example `<object>` `,` `or` <applet> ``.

__ OBJECT-SRC Policy Example__ `` object-src ‘self’; ‘’

media-src

Defines valid audio and video sources, for example <audio> , HTML5, <video> `` elements.

Example of MEDIA-SRC Policy media-src media.example.com;

frame-src

Defines valid fonts for loading frames. In CSP, Level 2 frame-src has been deprecated in favor of the child-srcdirective. CSP Level 3, has not been replaced with frame-src and will continue to postpone child-src if not present.

Example FRAME-SRC `` frame-src ‘self’; ‘’

sandbox

Enables a sandbox for the requested resource similar to the iframe sandbox attribute. The sandbox applies a policy of the same origin, avoids pop-ups, plug-ins and blocks the execution of scripts. You can keep the sandbox value empty to keep all restrictions in place, or add values: allow-forms allow-same-origin allow-scripts allow-popups, allow-modals, allow-orientation-lock, allow-pointer-lock, allow-presentation, allow-popups-to-escape-sandbox, eallow-top-navigation

Example of SANDBOX policy sandbox allow-forms allow-scripts;

report-uri

Instructs the browser to POST policy crash reports to this URI. You can also use Content-Security-Policy-Report-Only as the HTTP header name to instruct the browser to send reports only (does not block anything). This directive was discontinued at CSP Level 3 in favor of the report-todirective.

Example REPORT-URI report-uri / some-report-uri;

child-src

Defines valid fonts for web workers and nested browsing contexts loaded using elements such as <frame> '' and <iframe> ``

Example of FILHO-SRC Policy child-src 'self'

form-action

Defines valid fonts that can be used as an `<form>` HTML action.

Example action policy example `` form-action ‘self’; ‘’

frame-ancestors

Defines valid fonts for embedding the resource using `<frame> <iframe> <object> <embed> <applet>`. Setting this directive to ‘none’ should be roughly equivalent to X-Frame-Options: DENY FRAME-ANCESTORS__ EXAMPLE POLICY frame-ancestors ‘none’;

plugin-types

Defines valid MIME types for plug-ins called via <object> and `<embed>`. To load one, `` <applet> ‘’ you must specify application / x-java-applet.

PLUG-IN TYPES Policy Example `plugin-types application / pdf;`

base-uri

Defines a set of allowed URLs that can be used in the attribute of an HTML basetag.

URI-BASE Policy Example `` base-uri ‘self’; ‘’

report-to

Defines a report group name defined by an HTTP Response Report-Header. See the reporting API for more information.

Example Report for Policy report-to groupName;

worker-src

Restricts URLs that can be loaded as Worker, SharedWorker or ServiceWorker.

WORKER-SRC Policy Example worker-src 'none'; `

manifest-src

Restricts URLs where application manifests can be loaded.

Example of MANIFEST-SRC Policy manifest-src 'none'; `

prefetch-src

Defines valid sources for pre-fetching and pre-rendering of requests, for example, through the linktag with rel = “prefetch” or rel = “prerender”:

__ PREFETCH-SRC Policy Example__ prefetch-src 'none'`

navigate-to

Restricts the URLs to which the document can navigate by any means. For example, when a link is clicked, a form is submitted or window.location is called. If form-action is present, this directive will be ignored for submitting forms. Implementation status

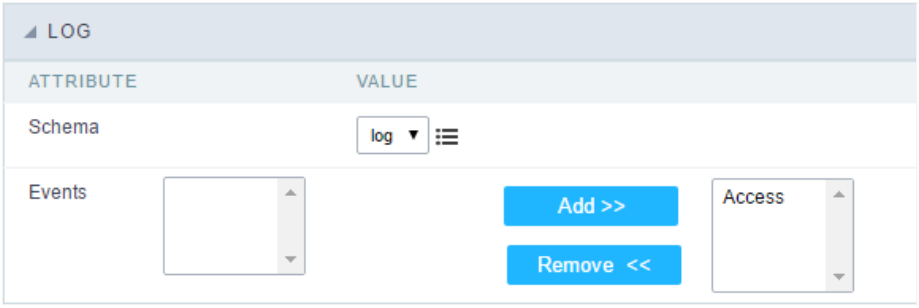
Navigation Policy Example `navigate-to example.com`

Responsive Menu Log

Log

This interface allows you to define a Log schema to the app. The Log scheme tracks what the end-users are doing in the app. Those are the events it can tracks: Access, Insert, Update, and delete.

To learn how to create a Log schema see here.



Application Log configuration

Schema Select an existing log schema for the application.

Events You can select just specific events to save in the log.

Responsive Menu Events Overview

Most Scriptcase applications, with the exception of Tabs and Dashboard applications, have events that can be used to create business rules, validations and changes to the application's CSS, among other things.

When events are executed?

Each event is executed at a certain moment in the application and has an execution order, for this reason, they have different functions within the creation of a business rule and validations.

They can be triggered automatically, as in the case of the **onApplicationInit** and **onScriptInit** events, where they are executed when the application starts. Also, they can be triggered through user interaction, as with the **onNavigate** and **onValidate** events, which are executed when browsing the application or submitting a form, respectively.

> For information about the timing of events execution or other information, access the documentation of the desired event.

how they work?

By default, the events already have handle for opening `<?php` and closing `?>` of PHP tags, so they are ready to receive PHP code and macros from Scriptcase.

However, it is possible to use other languages in the event, such as **JQuery**, **JavaScript**, **CSS** and **HTML**. For this, it is necessary to close the PHP tag `?>` of the event, so that the code can be added.

Remembering that it is necessary to reopen the event's PHP tag `<?php`, otherwise the application will not run correctly.

> When closing the PHP tag in a Scriptcase event, the executed language is HTML.

Example of using other languages:

onApplicationInit - Responsive Menu

onApplicarionInit

This event occurred before the application execute the SQL, and execute only once. Is used to do verification of variables, and security verification.

onExecute - Responsive Menu

onExecute

This event works only in the "menu" applications and are executed when an application is called through a link of the menu. Normally are used to the decision taking before the application execution named sc_script_name.

onLoad - Responsive Menu

onLoad

This event occurs when loading a horizontal / vertical menu. In this event we can run some security policy, for example if there is an attempt to access the menu directly, without first going through a login form, we can test a session variable defined in the login form, redirect the flow to the initial application. if(!isset([var_id_user]) or empty([var_id_user])) { sc_redirect("login.php"); }

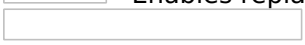
Code Editor

Code Editor

Scriptcase has Events to allow the developers to customize the application code. By using the events, the developers can program custom actions at a specific execution time (e.g. After a record is inserted, when it is loaded, upon submitting a Form, ...) and for a specific application type. In the events area you can use global and local variables, JavaScript, CSS, HTML, PHP codes and also Scriptcase macros.

Scriptcase code editor is used within events and has a series of shortcuts to help in the applications development.

At the top section of the editor, there are some options that allows us to change the editor theme or expand it, for example. These options can also be activated using the hotkeys.

- - Increases the area occupied by your code editor.
- - Expand the sides of the code editor.
- - Activates the search in the code editor.
- - Enables replace in the code editor.
-  - Defines the code editor theme.

The hotkeys are available in all browsers compatible with Scriptcase.

> Click Here to view the Scriptcase hotkeys documentation.

Responsive Menu Settings

Settings

Application Code

Informs the current name of the application and the version in which it was created

Language

Defines a specific language for the application, overriding the default language defined in the project. If no language is selected, the application will inherit the project's default language.

> The languages listed in the option are defined in the project properties

Editing by Project

By disabling this option, only the user who created the application will have access to its settings.

No other project users will be able to access the application.

In the image below, the menu application has the edit by project option disabled.

Session Location

Defines if the language of the applications will be stored

Charset

Defines an application-specific charset.

If no value is selected, the application inherits the project's default charset.

> Ideally, this charset option, within the application's settings, does not have a defined value. Different usage of your base charset can cause problems saving information in your tables.

Schemes in Session

When active, this option allows the theme stored in the session to be used by the application.

For example, by selecting theme x at login, all applications in the project will use the selected theme.

If the option is unchecked, the application remains with its original theme.

Folder

Folder where the application is stored in Project Explorer.

Description

Displays the description of the application.

You can edit or add a description also in the Project Explorer interface.

Extra Application Icons

It stores the images that will be used in the application through codes in the events, causing these images to be sent along with the application at the time of publication.

Only images used in codes, such as creating a dynamic menu with icons or creating a bill of exchange using

a blank, need to be added.

Images added in HTML Image fields or in the application header, for example, are already sent with the applications.

The added images will be stored in this directory: **../_lib/img/**

How to use the images

When inserting the image in the image manager, its name is changed according to the **scope** (public, project or user) and the **storage directory** (background, button, icon, menu or general) following the following pattern.

Scope: It refers to the level of access to the file within Scriptcase, in the development environment.

- Scriptcase__ - Contains preexisting images in the tool, where it is not possible to upload new images.
- Public__ - Images accessible for all projects.
- Project__ - Images accessible only in the project in which it was inserted.
- User__ - Images accessible only to the user who uploaded the image.

Each scope adds a prefix to the image name.

- Public - **sys__NM__**
- Project - **grp__NM__**
- User - **usr__NM__**

Storage Directory: Directory where the image will be added. Each directory adds one more prefix to the image name.

- background images - **bg__NM__**
- button images - **btn__NM__**
- general images - **img__NM__**
- icons - **ico__NM__**
- menu images - **menu_img__NM__**

This way, an image called _001.jpg, added in the public scope and in the icon directory will be named **sys__NM__ico__NM__001.jpg**

HelpCase Link

Associates HelpCase files to your application.

This configuration can be done expressly in the HelpCase settings, where it is possible to change this option for all project applications. See how by clicking here.

The HelpCase button must be selected in the toolbar to be able to access the selected page.

Notifications Settings

Use SweetAlert

Activate SweetAlert in the application, replacing the “confirm” and “alert” of the browser.

Confirm Example with SweetAlert Enabled

Confirm Example with SweetAlert Disabled

SweetAlert position using Toast

Sets the display position of the message window using toast.

Global Variables

Global Variables

This screen shows all global variables used in the application

The global variable is an external parameter required for running the application. The application can include global variables in the WHERE clause, field definitions and names, event programming, and so on.

You can define global variables by using square brackets ([variable]). You can pass the parameters to the application through one of the methods: Session, Post, and Get

> **IMPORTANT:** if you need to use Database Identifiers in your tables, we recommend using the double quotation marks(**“**) instead of the brackets(**[ ]**) because of the conflict with the of Global Variables syntax{:target=”_blank”} Scriptcase. **Using the brackets as database identifiers may lead to problems in the Scriptcase applications**.

Global variables Interface.

Example:

```
Select CustomerID, CustomerName from Customers WHERE CustomerID = '[v_customerid]'
```

> We do not recommend using session variables (global variables) with the same name as the table fields.

In this case, the variable **v_customerid** is displaying in the global variables configuration.

Global variables configuration Interface.

Attribute

It shows the variable’s name in the application.

Value

Defines the behavior of the variables, divided into three blocks, they are:

Scope

Defines how the application receives the variable. A variable defined as the POST method in the app does not receive value if it comes with the GET method. Those are the methods available:

- **Session**: Defines that the variable must be created as a PHP session variable by another application.
- **GET**: Defines that the variable must be passed by the URL, that is, visible by the browser.
- **POST**: Defines that the variable must be passed through the POST method of an HTML form.

Settings

- **Optional**: Check it to don’t validate the variable at runtime.

Type (In / Out)

Defines whether the variable comes from another application (In), or it is starting in the current app (Out).

Description

Indicates where the application is using the variable.

Import Menu

Import Menu

Allows you to import items and events from an old menu application to the new menu application.

Attributes

ScriptCase has incorporated the concept of Object Oriented programming, using attributes, resources, methods and libraries. It is possible to create your own business rules in applications, and by using these concepts you can reap huge rewards in terms of better organization and improved development.

Attributes Settings

The attributes are variables that has the global scope declared in the application. An attribute that can be referenced in all the methods and the application’s events.

Attributes configuration Interface.

Attribute Name : This field allows to inform the attributes name, after typing the name just click on **include**. In case you want to **update** click on the attributes name (located on the buttons’ right side), do the necessary modifications and click on update. In case it is necessary to **delete** an attribute, it is necessary to do same process as the update, but when selecting the attribute, it is necessary to click on the delete button. The **clear** button, as the name suggests, it clears the list of all the attributes listed on the right.

Internal libraries

Internal libraries

To use the internal libraries just select the correspondent (Project Libraries for example). This process allows the methods existing in the library, to be visible in all the application’s Events and Methods.

See how to manage the libraries by clicking here.

Internal Libraries managment Interface

PHP Methods

PHP Methods

Methods are function or procedures declared by the developer, that helps when applying the routines. Using methods in the applications allows to reuse your code throughout the application, optimizing the development experience.

Creating a new method

- Give the method a name and click Create. As in the image below.

  PHP methods are incorporated into the same class, so it is not possible to use two functions with the same name even when used in different methods.

- Methods can receive parameters.

- Add the amount of variables:

- Defining the variables:

- **Name** : Type in the variable’s name.
- **Type** : Selecting the type of variables: For Value or For Reference.
- **Value Standard** : The parameter’s value used to initialize when calling the method.

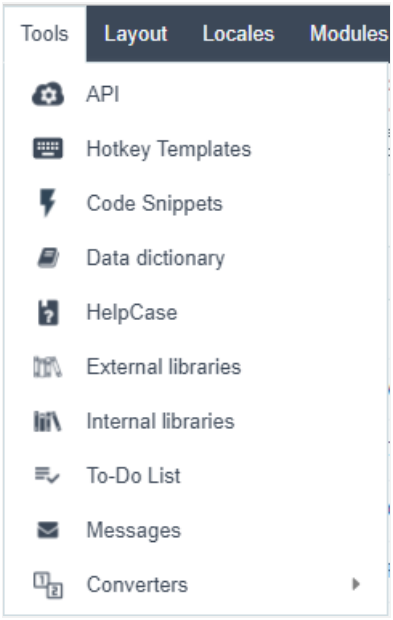
  References allows to create a second name for a variable that you can use to read and modify the variable’s original information.

- Editing a parameter:

- : Checking all the parameters in the parameter list.
- : Uncheck all the parameters in the parameter list.
- : Edit the selected parameter of the list.
- : Deletes the selected variable of the list.

Scriptcase Tools Overview

Scriptcase offers various tools that help you out on developing systems to optimize the time of development, making it easier to create and maintain your applications within Scriptcase. In this tutorial, you learn what the environment tools are.



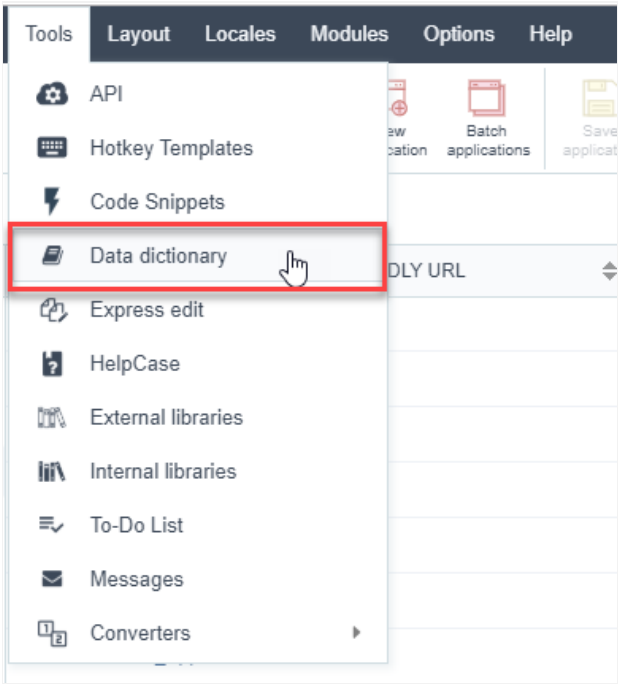
- **Data Dictionary** - You can use it to standardize the tables fields. It helps you to manage the language variables so you can translate the applications created within Scriptcase as well as model the fields to create the applications faster.
- **External Libraries** - This tool offers the option to include external libraries to your project like JavaScript, CSS, HTML, and other libraries. It allows you to integrate the various libraries to your applications.
- **Internal Libraries** - This tool offers the option to create PHP functions to be used in the applications.
- **To-Do List** - Allows you to create a To-Do List and assign Tasks to your developers.
- **Messages** - Allows the developers to exchange messages between themselves in the same ScriptCase.
- **Converters** - This tool offers the option to convert your projects from previous versions of ScriptCase to the current version.
- **Database Import** - This tool offers the option to import tables in the formats (XLS, CSV, and ACCESS) for the following databases: MySQL, PostgreSQL, SQLite, and SQL Server
- **SQL Builder** - This tool is an interface that provides some options to create a **SELECT** query to be used in your applications.
- **Database Builder** - This tool is an interface to manage your database directly from ScriptCase, without needing to use any other software. It supports the Databases: MySQL, SqlServer (not ODBC), SQLite, Oracle, PostgreSQL.

Data Dictionary

In this article, you learn how to use the Data Dictionary. It is a tool used to assist you with the application translation as well as to standardize the database fields. With the data dictionary, you can create a data repository and associate it to one or more tables.

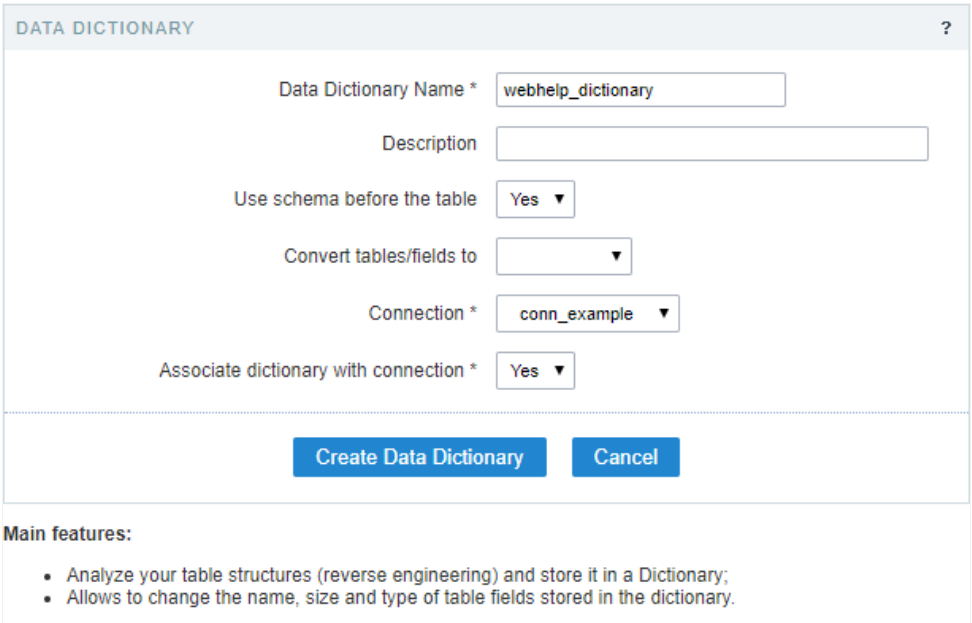
WATCH TUTORIAL

You can access the data dictionary from the main menu **Tools > Data Dictionary**.



Creating a Data Dictionary

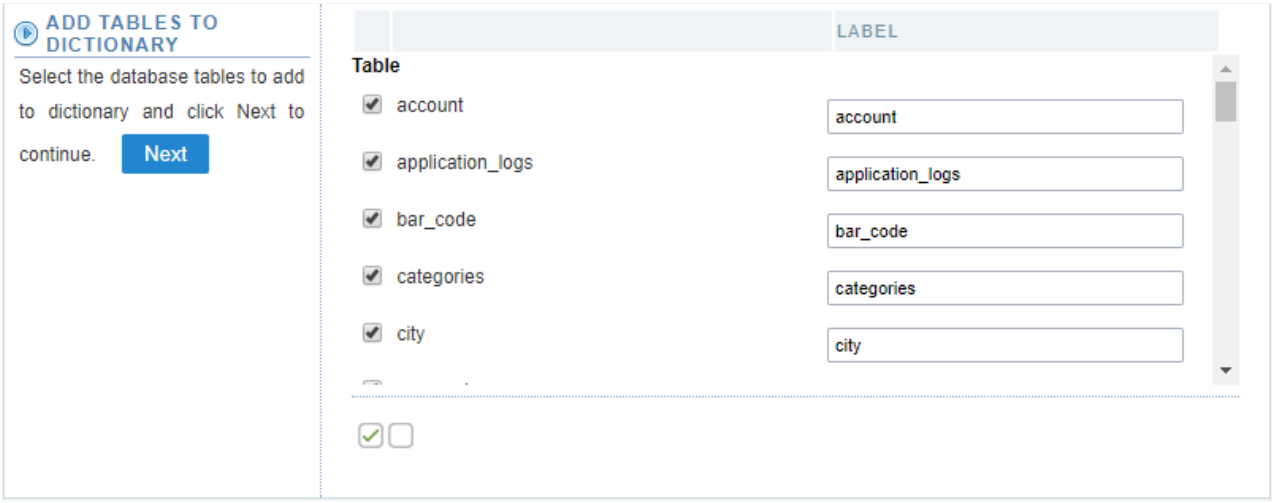
Inform the necessary data for the creation of the repository. The files with an asterisk are required.



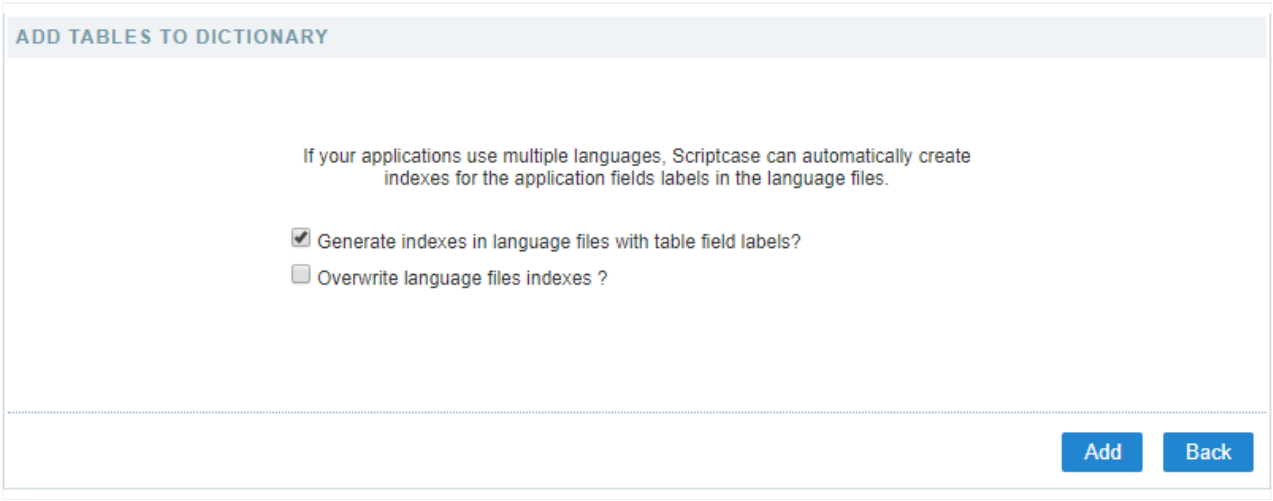
The option **Associate dictionary with connection**, define a connection to associate with the repository data, allowing that the new applications, created using the tables associated with the repository, have indexes in the language files with the field labels from the tables.

Add tables to the Data Repository

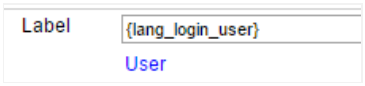
Here we need to select the tables to add to the repository. After selecting the desired tables, click on next.



You should choose whether or not to create the indexes (langs) on the field labels of the existing applications of the project.

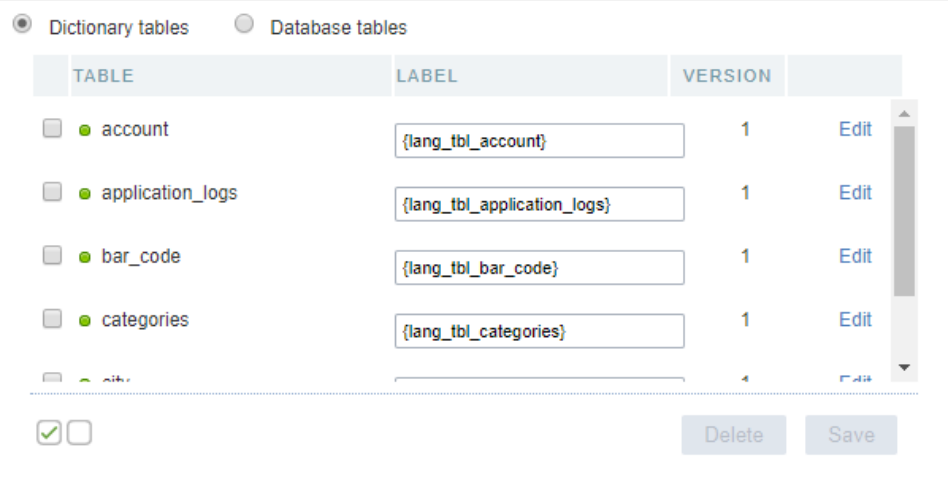


- Generate the indexes in the language files of the table fields - On this option, you replace the field labels with the indexes that can be edited in one place when accessing **Locales > Application Language**.

An example of an index created by the Data Dictionary: 

- Overwriting the indexes in the language files - On this option, you replace the existing indexes with new data based on the database.

Next, we can view a list of tables added to the repository data, where we can view a couple of information about these tables, as the name of the table of the database and the index created for each table, on the label field, and the version of the Data Dictionary of each field.



The tables that not added can be viewed when accessing the **Dictionary tables**; the process of including these tables to the repository of data is the same.

When doing a modification to the structure of the database tables, the same should be updated using the option **SYNCHRONIZE DICTIONARY**.

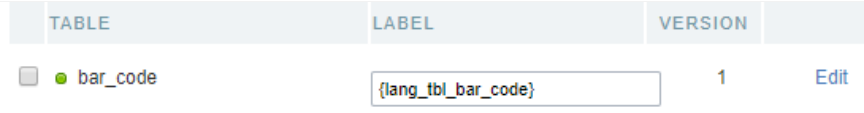
Table status

Next to the name of the tables, there's an icon that indicates the status of the tables.

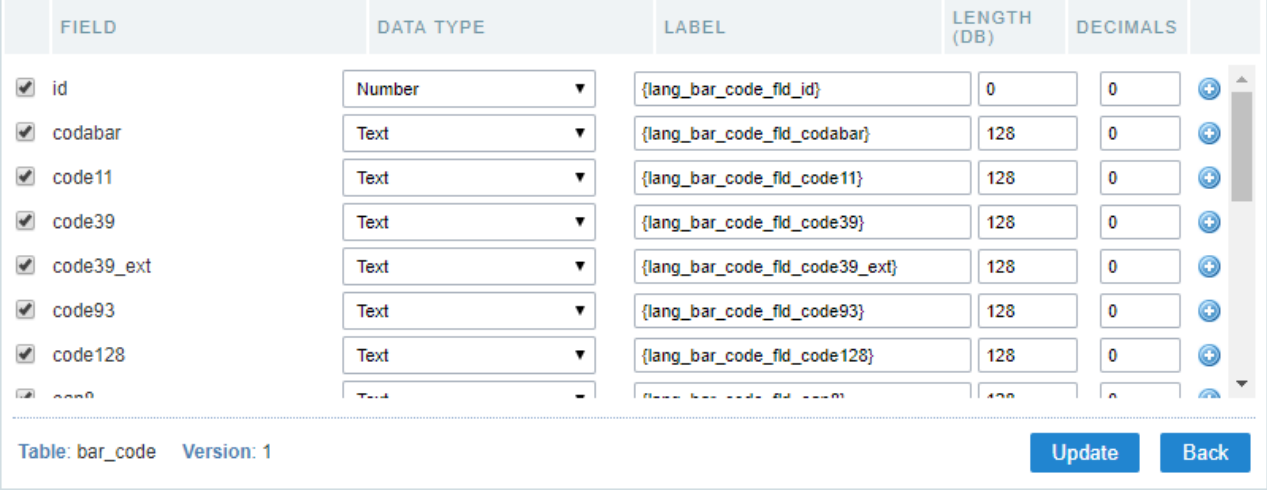
- Green - Indicates that the tables and the repository are updated.
- Red - Indicates that the tables don't exist in the database. In this case, you should delete it from the repository.
- Yellow - Indicates that the tables are not updated because of some modification in its structure. In these cases, you should click o the link **edit** referring to the table and click on update, so that the data of the repository could update.

Edit Tables

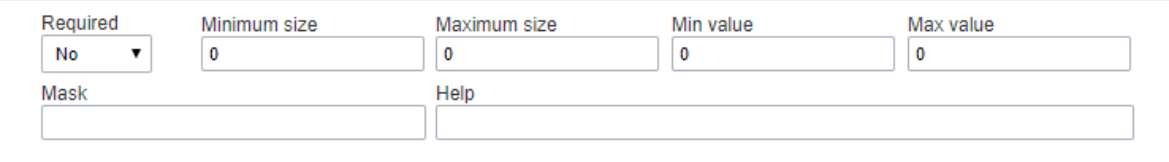
You can find it on the right side, in the list of tables.



This option allows you to edit the fields from the chosen table, like the data types and the indexes created by the repository, for example.

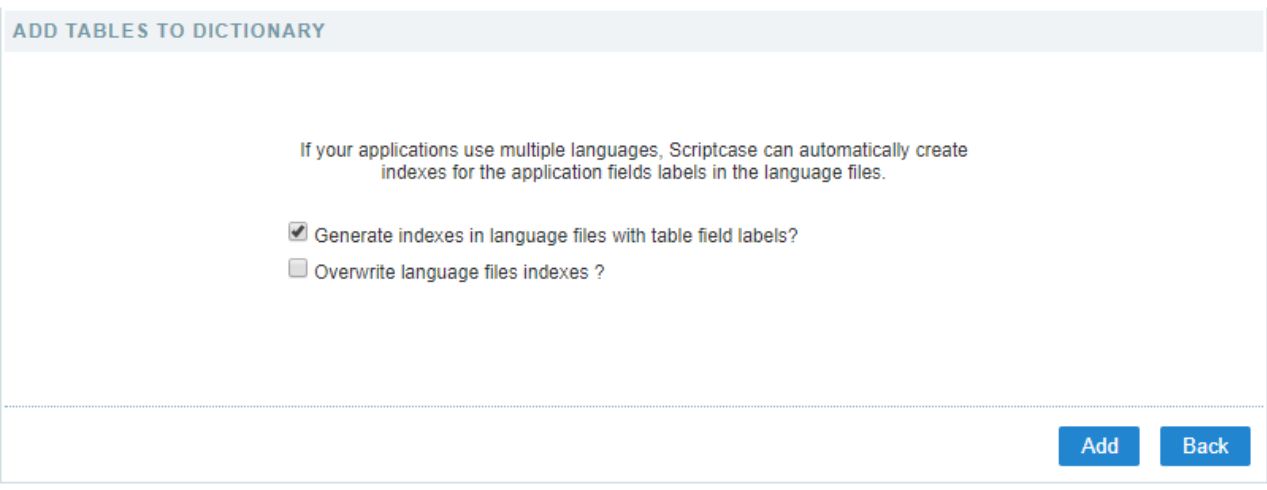


Other properties of the fields can be accessed when clicking the plus icon on the right side of the field list.



Syncing Dictionary

This option is activated when selecting one or more tables; on this option, you update the data dictionary according to the database.



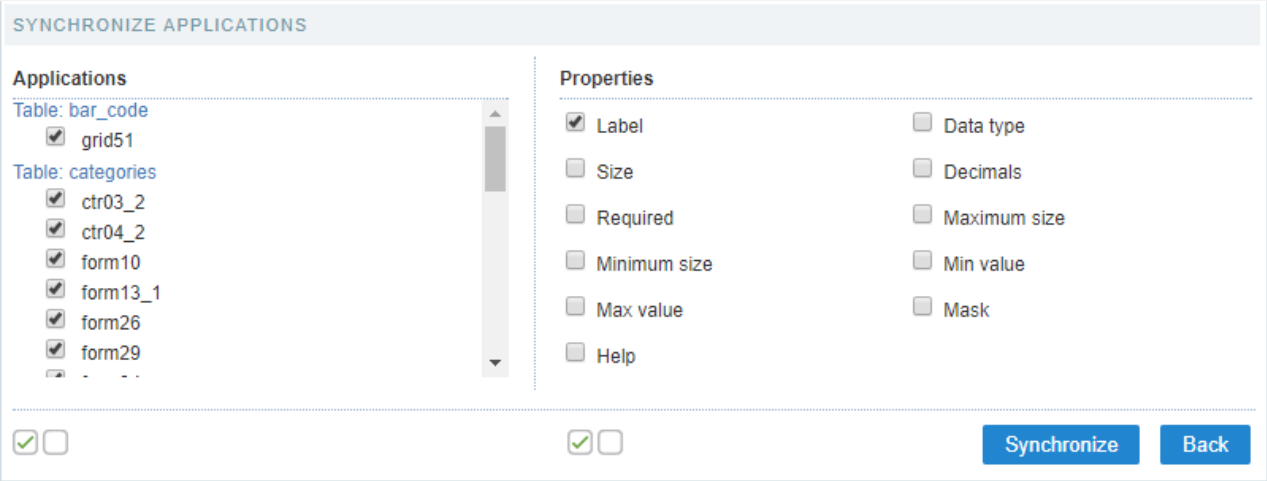
When synchronizing the tables, a yellow icon is displayed on the side of the updated tables, **See how to remove it**. In case of changing the structure of the tables, then you need to re-sync them again.

Synchronizing Applications

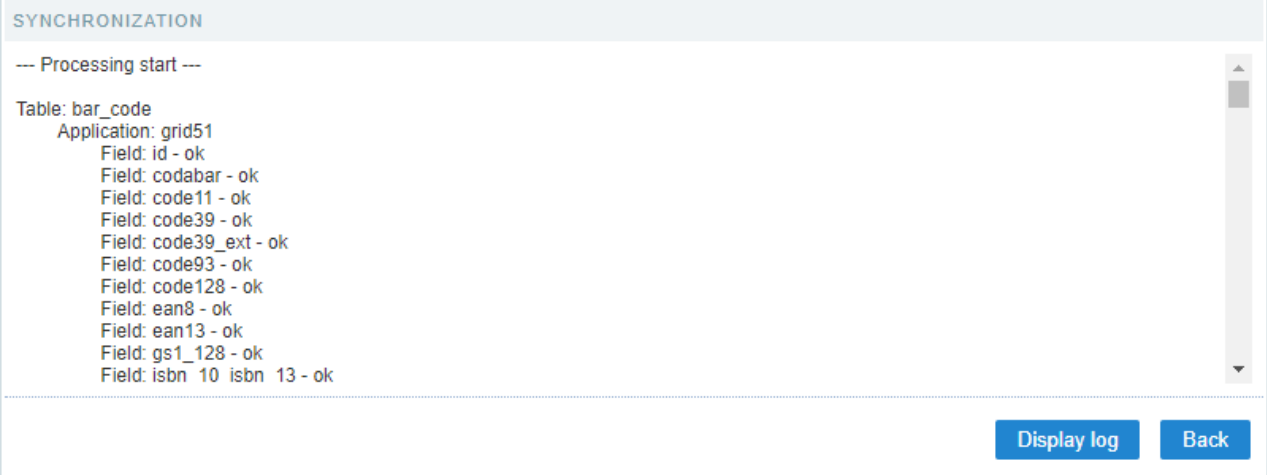
After doing all the desired configurations on the tables and fields, you need to sync the changes to the existing applications, so that the changes can be applied.

First, you need to select the tables that are synchronized.

Next, you can define what happens and which applications suffer the modifications — a list of applications that received the changes based on the tables that they used.



When clicking on synchronizing, when finishing the process, you see a summary of the procedure done.



Clicking to show the log, you can view all the modifications done, where it displays the previous state (blue) and the result of the modification (red)

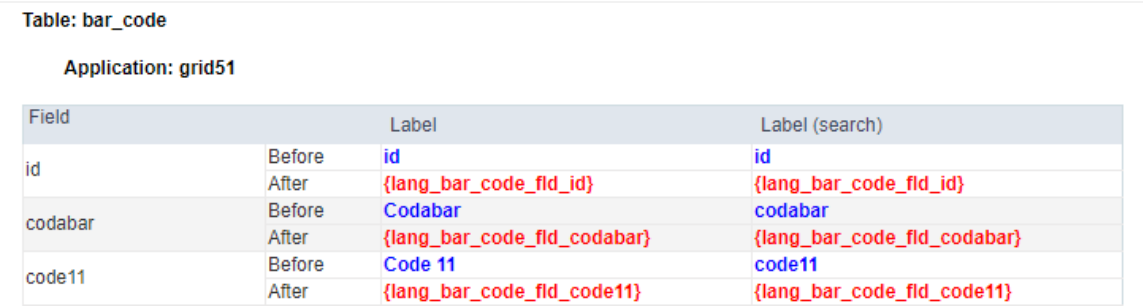


Table: bar_code

Application: grid51

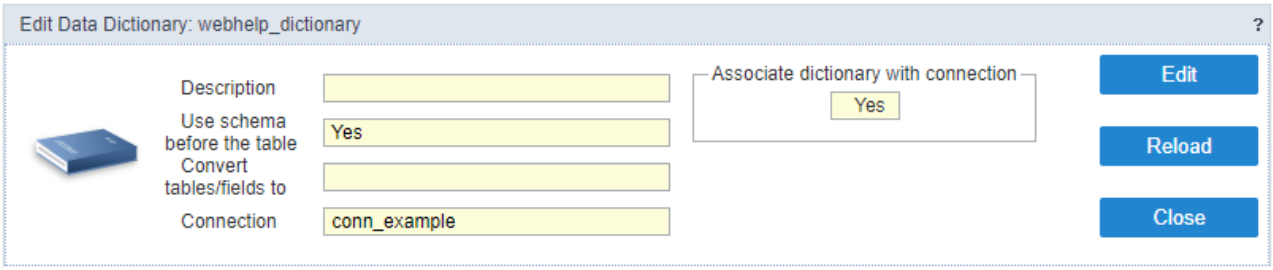
Field		Label	Label (search)
id	Before	id	id
	After	{lang_bar_code_fld_id}	{lang_bar_code_fld_id}
codabar	Before	Codabar	codabar
	After	{lang_bar_code_fld_codabar}	{lang_bar_code_fld_codabar}
code11	Before	Code 11	code11
	After	{lang_bar_code_fld_code11}	{lang_bar_code_fld_code11}

Clicking on **Back** You return to the Page of the definition of modifications.

To finish the process, click on close.

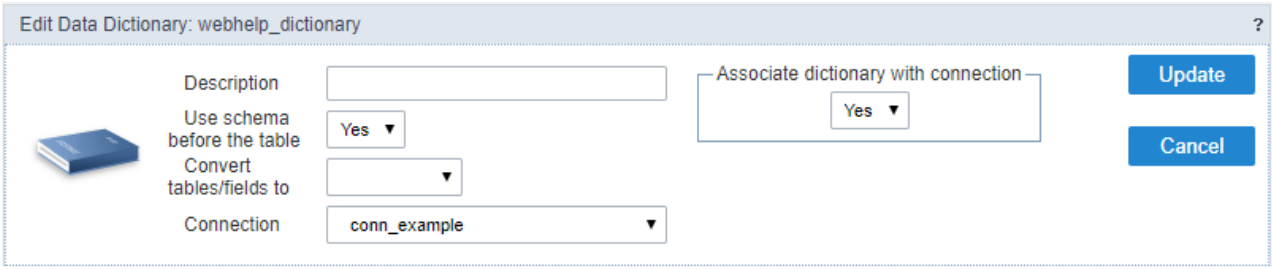
Editing the Data Dictionary Properties

On the top part, we can view the dictionary properties, informed when created.



You have three possible actions, Edit, Reload, and Close.

- Edit - Allows you to change the properties data of the dictionary, like the modification of the connection used to associated with a repository, for example.



- Reload - Updates all the information of the Data Dictionary, like the status of the table, for example.
- Close - Closes the current data dictionary, and you go to the list of the dictionary.

Editing Indexes

To edit the created indexes in the repository, access **Locales > Application Language**.

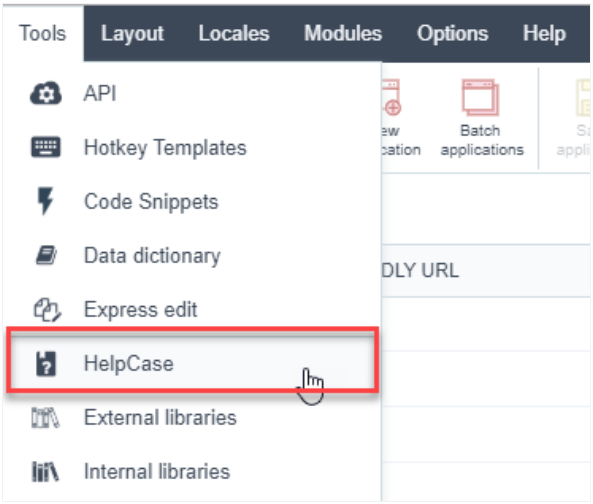
On the side panel, it is possible to view all the lang from ScriptCase. The langs referring to the data dictionary is in the folder **Project Messages**.

- Application Language
- Regional settings

HelpCase

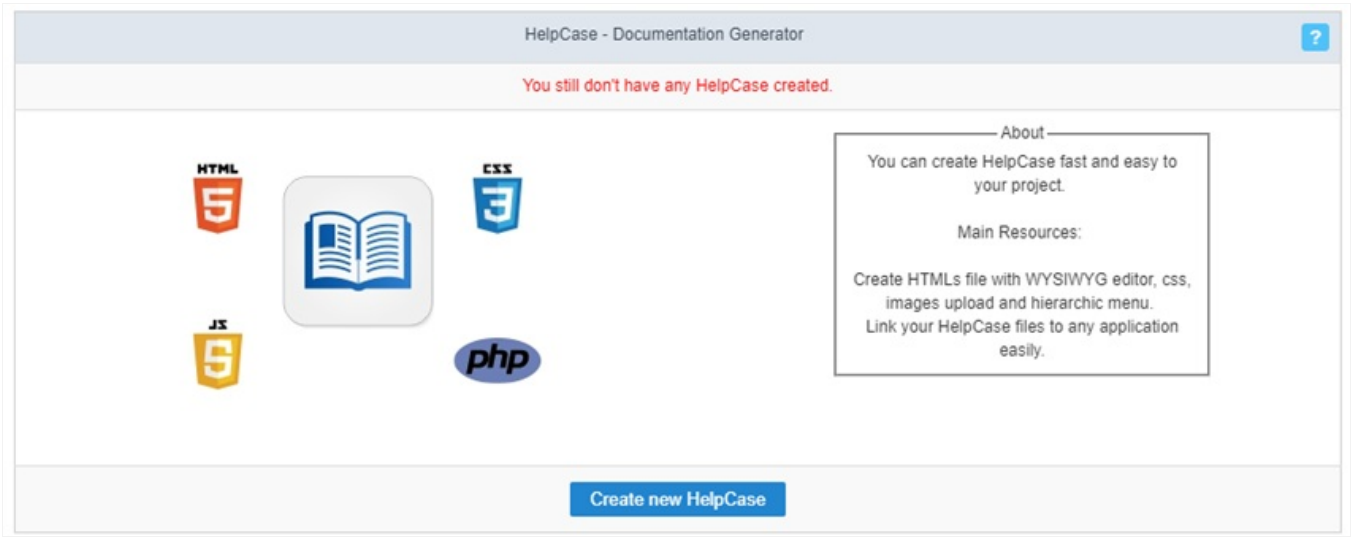
In this article, you learn how to create the documentation for your project using this integrated tool: HelpCase.

You find the HelpCase from the menu at **Tools > HelpCase**.

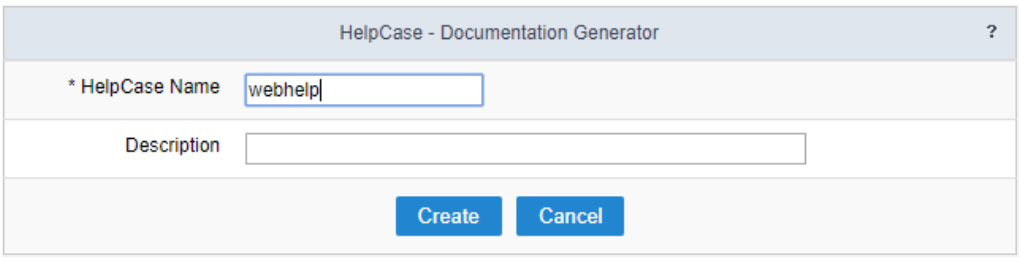


How to Create a HelpCase

The HelpCase creates HTML files with WYSIWYG editor, CSS, images upload, and hierarchic menu. You can also link HelpCase files to the applications you create within Scriptcase.



The first step to creating a document using the HelpCase is to enter a name and description.



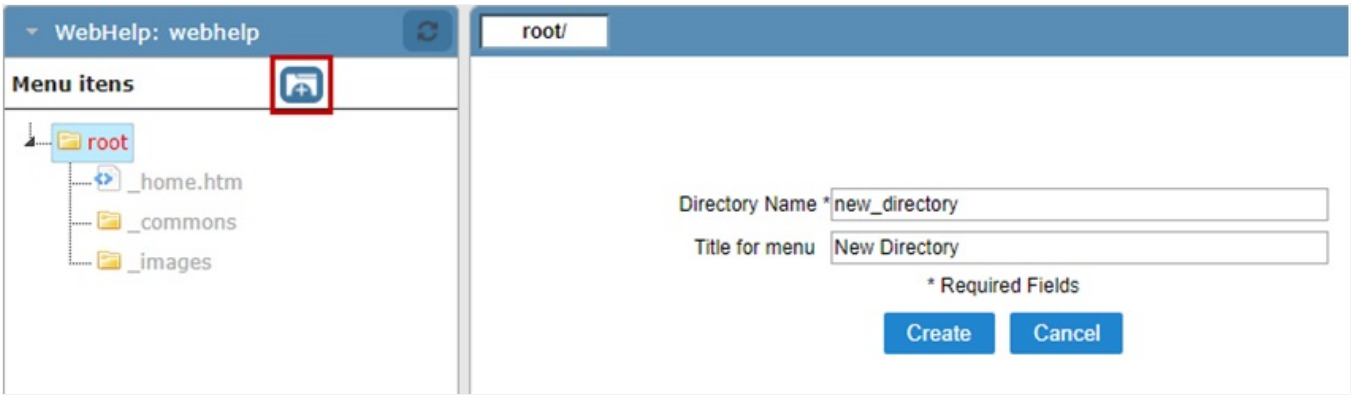
After creating the HelpCase, you need to generate the content of the documented manual of the functionalities and description

of the project.

On the left side, in the block 'WebHelp,' you see the folder structure. Still, in this block, we can create the folders of the documentation.

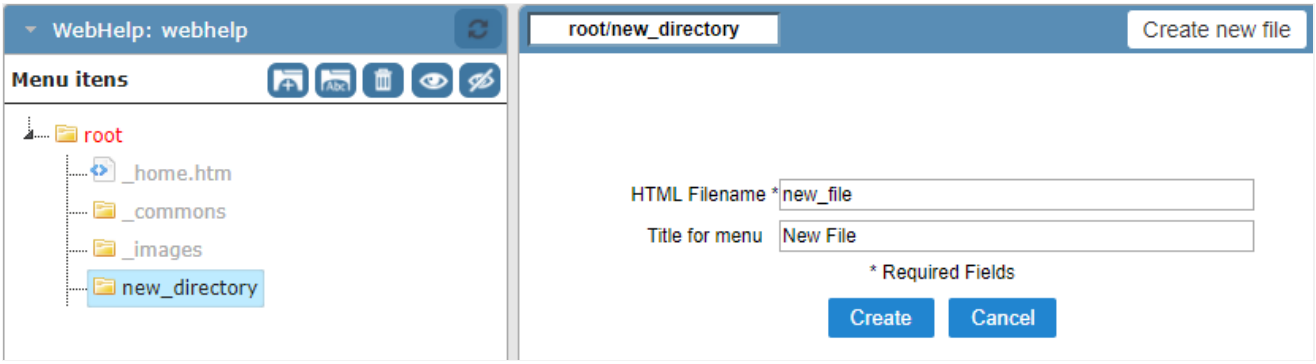
Select the root or any other folder on the side panel and click on the folder icon to create a folder in the select directory.

Inform the directory name (Special characters are not acceptable) and the display name in the documentation menu.



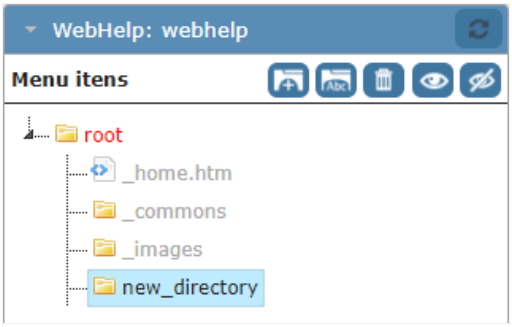
To create .html files, where you place your content, you need to select the directory where you place the files.

Inform the name of the file (Special characters are not acceptable) and the display name in the documentation menu.



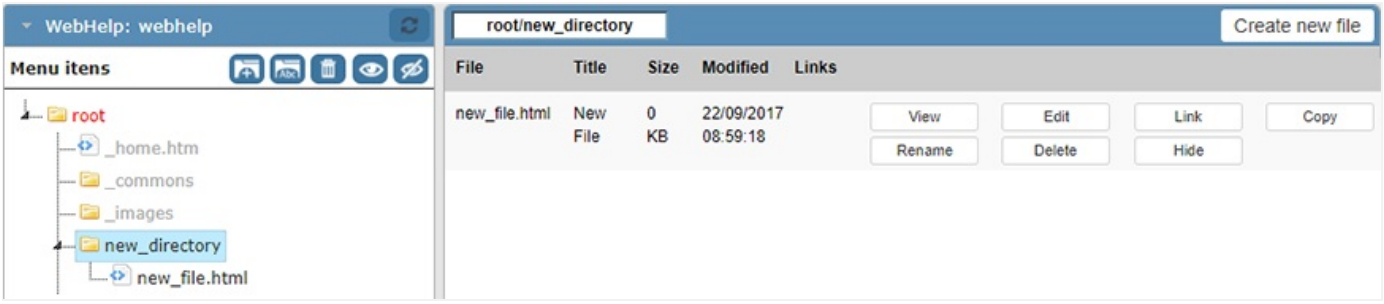
Editing the HelpCase

When you select the folder, you see some options.



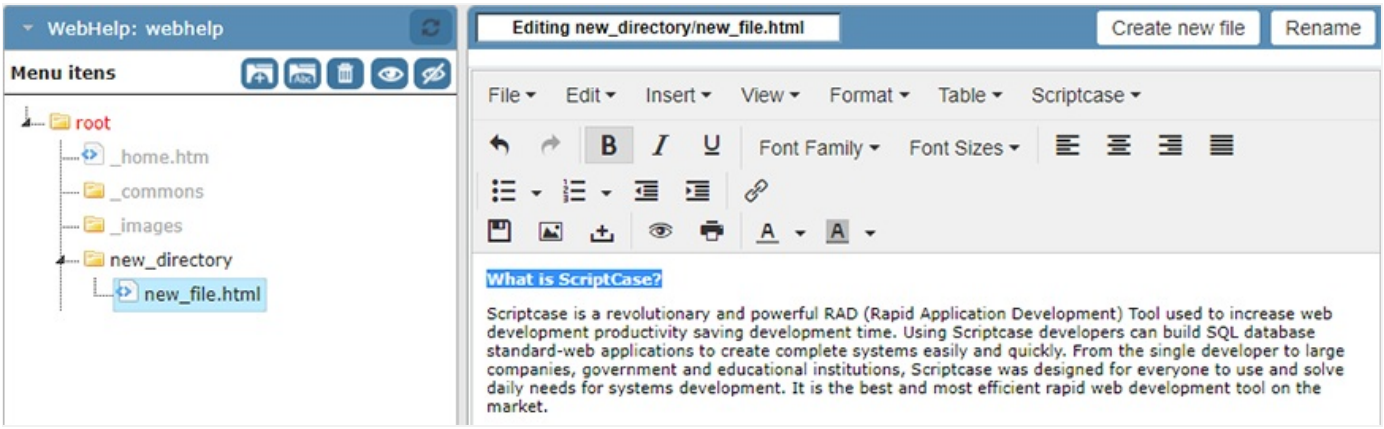
- **Icon (Folder with letters)** - Allows you to rename the archives and the display name for the menu.
- **Icon (Recycle Bin)** - Deletes the directory with all its content.
- **Icon (Eye)** - View the directory in the documentation menu.
- **Icon (Slashed Eye)** - Disable the view of the directory in the documentation menu.

After created, the files are displayed on the list when clicking on the created directory. Details about the files can be viewed, like the display title on the menu and its actual size, also other options like rename and delete.



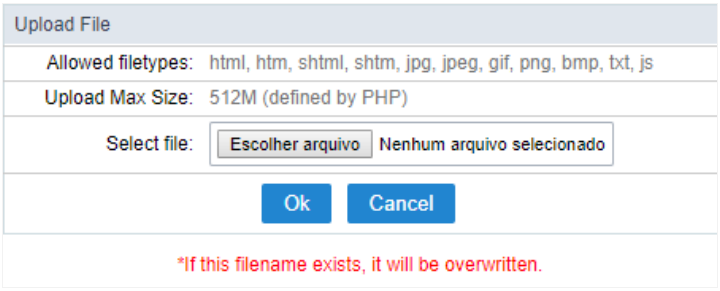
- **View** - View the content of the file.
- **Edit** - Open the file to place your content in it.
- **Link** - Create a link between the file and the applications selected.
- **Copy** - Creates a copy of the file.
- **Rename** - Rename the file and it is display menu.
- **Delete** - Deletes the chosen file.

After the creation or clicking on edit a file, you can insert the content that will be displayed for the end user.



Upload

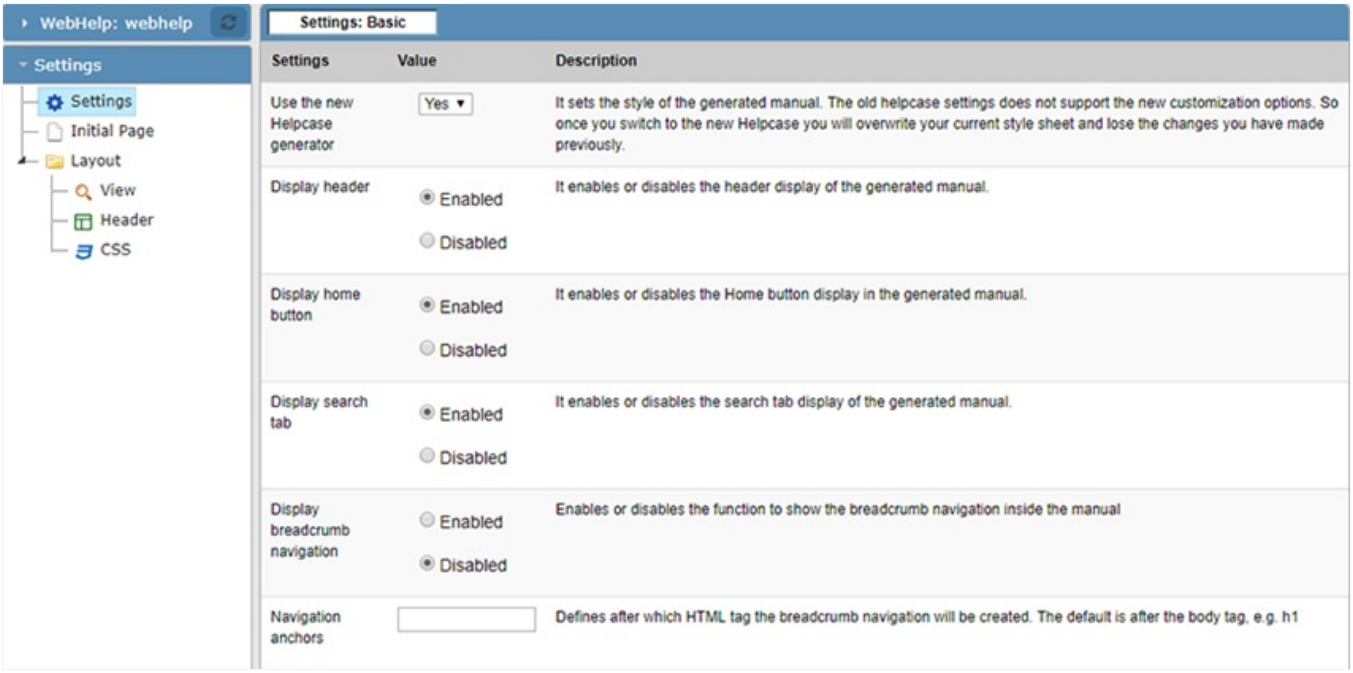
You can upload already created content or images to use in the manual. The upload of these files, can be done by clicking on the upload button, on the upload page you will see the supported formats and size limit of the files.



General Settings

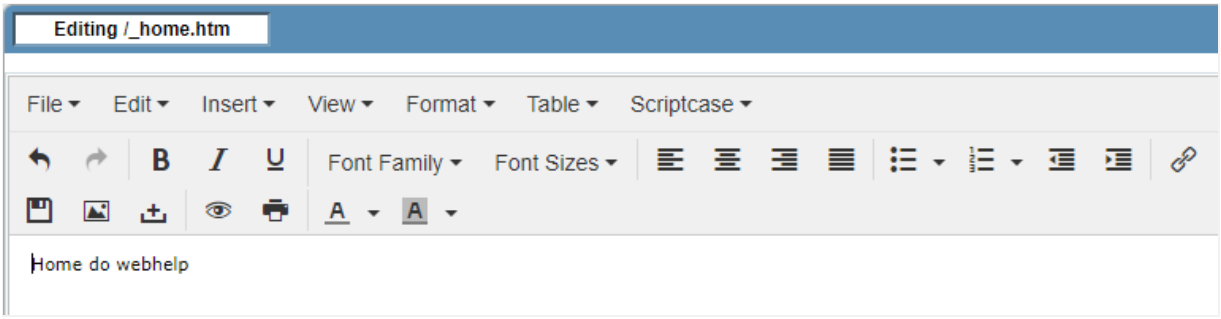
Settings

Allows you to setup some things, like the header display and the search tab.



Initial Page

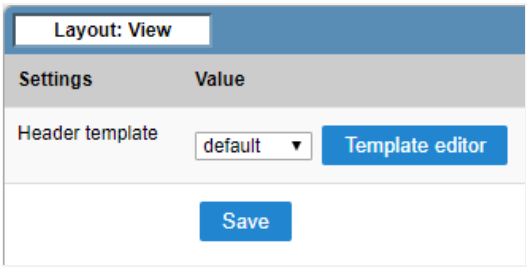
Allows to write content of the initial page of the documentation. The modification can be done also when selecting the `_home.htm` file on the files panel.



View

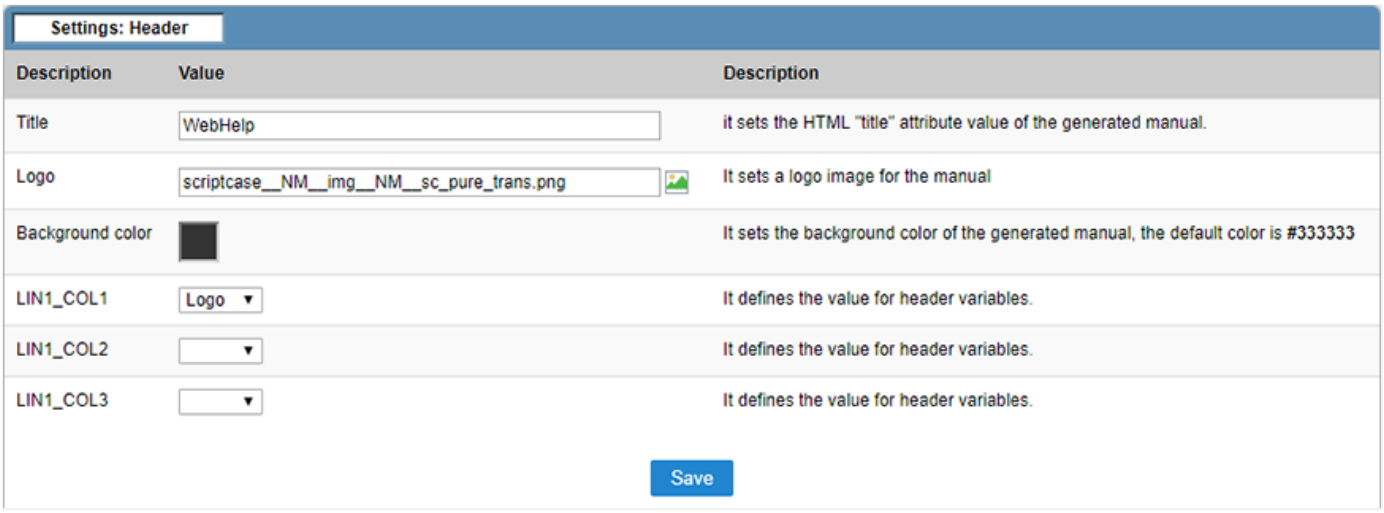
Define a template to be used in the header of the manual, it is possible to edit or create new templates in the editor.

This editor can be accessed from the menu **Layout > HTML Templates**



Header

Allows to configure what will be displayed in the header.



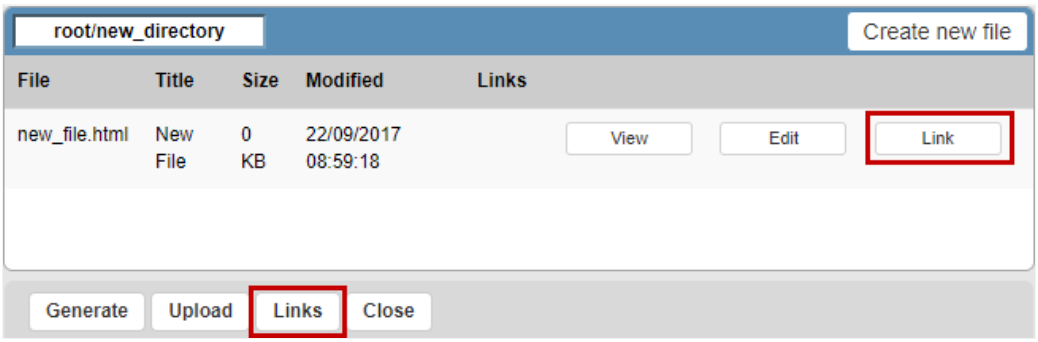
CSS

Allows to modify the HelpCase CSS.

After creating all the content and the configuration of the layout, you need to associate the create files with the applications, so that the end user can have access to the information that he needs in a quick way.

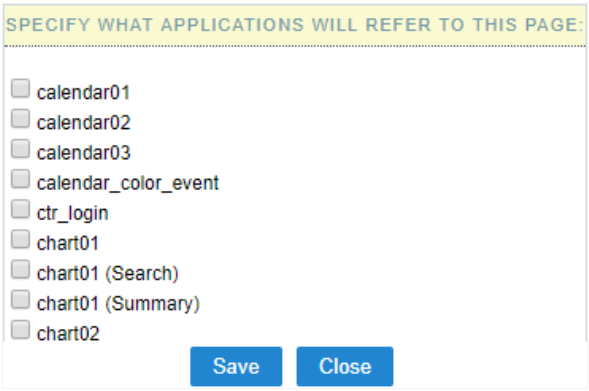
Associating Applications

You can start associating all the applications at once by clicking on **links**, or clicking on **link**, where you can associate only the chosen file to various applications.



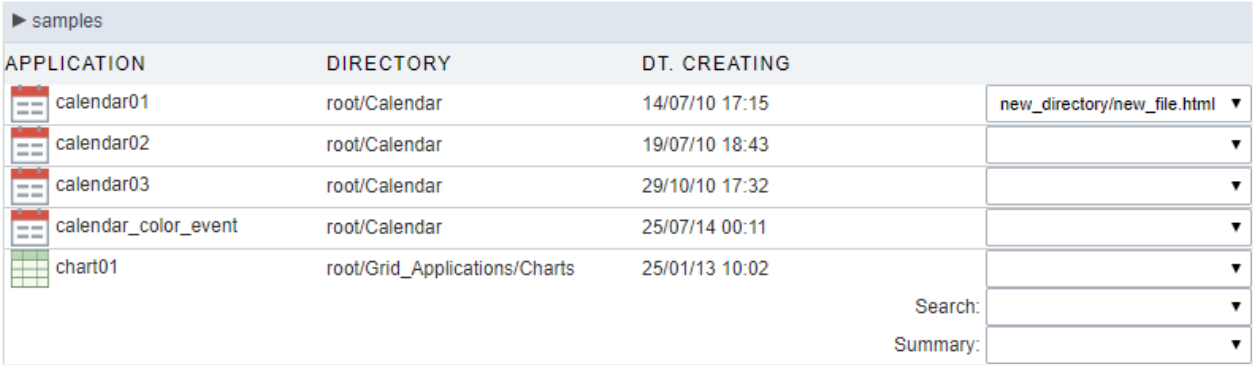
Links

On this option, you will see a list of the applications, where you should select the applications that will be associated.



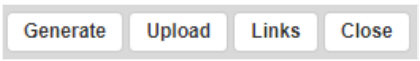
Link

Clicking on **link**, you will be presented a list of all the applications, in this case you need to select the files that will be associated for each application.

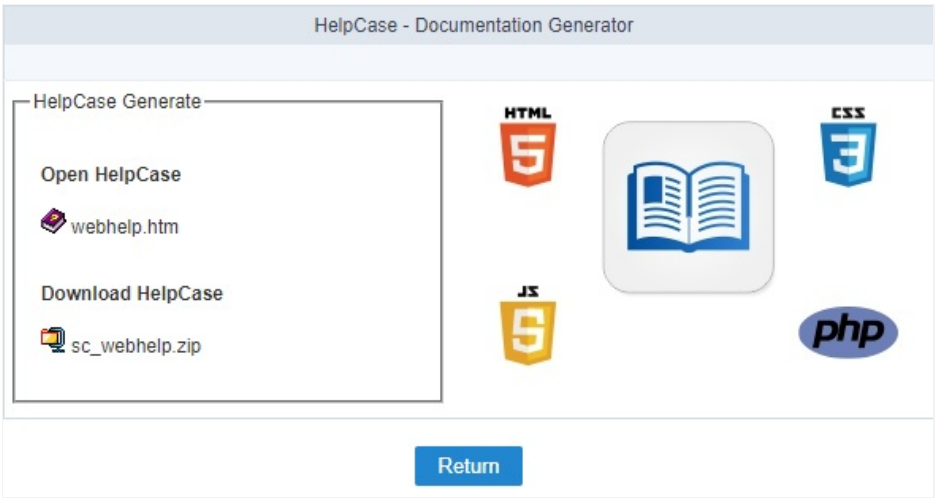


Generating Manual

Generating the manual is the last step of its creation, after doing all the configurations, click on **generate**.



After generating the manual, you can download or access it.



For you to have access to the manual from the applications, you need to click on the button **WebHelp** on the toolbar of the applications that have a link to the manual.

External Libraries

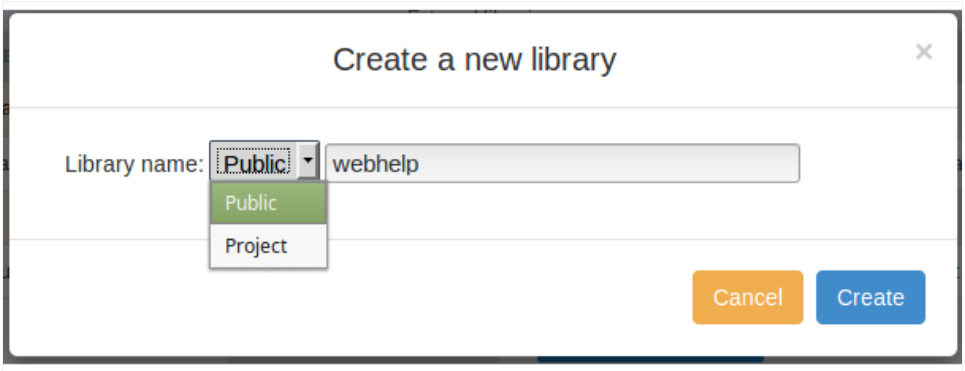
In this article, you learn how to use the External Library. This feature allows us to import complete libraries and create files of any extension (.JS, .CSS, .PHP...) to use in your projects, making it easier to reuse some of the source code and to bring external codes libraries to implement features to your applications.

You can find this option within the menu **Tools > External Libraries**, where you can manage the libraries of your project.

To use this functionality, you need to use the macro, **sc_url_library()** and the **sc_include_library()** (Only for PHP files)to import the library to the applications.

Creating a new library

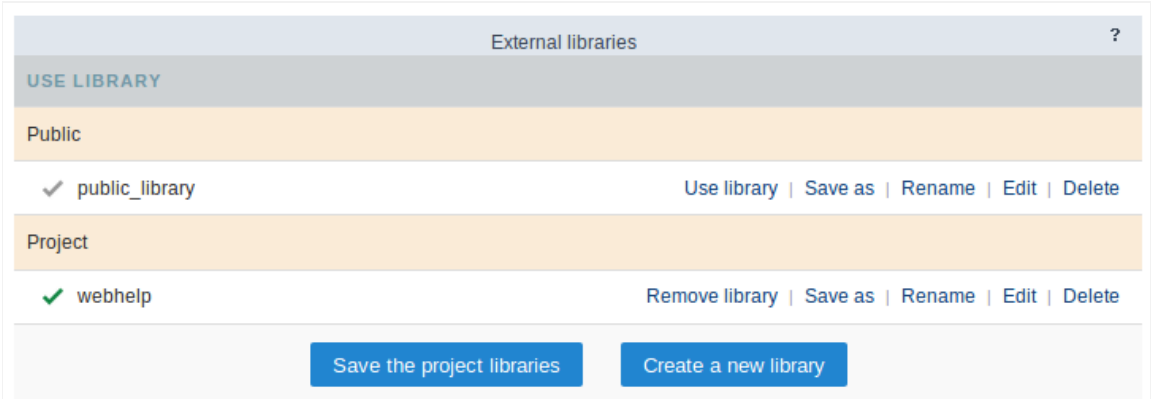
To create a library, click on **Create a new library**. Next, inform the library name and the access level of the library.



- **Public** - Available for all the ScriptCase projects.
- **Project** - Available only within the current project.

Managing the Libraries

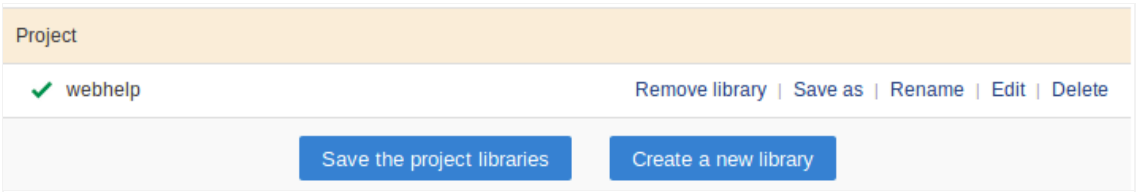
All the public and project libraries are listed here. This page allows you to select the libraries that is part of the project, delete the unnecessary libraries, rename, or edit them.



Using a Library

This option defines a library used in the current project. The enabled libraries, for this project, can be differentiated by the checkmark next to the name of the library.

When you select a library to the project, the icon changes color when you click on the button “Save the library to the project” and enabled, you need to click on the button to save the modifications.



Rename

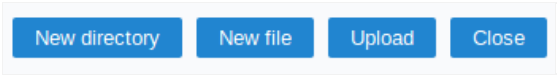
To rename a library, just click on “rename” and then inform the new name for the library.

To rename a library, you will need to change all the includes of the library in the events of the applications.

Editing a Library

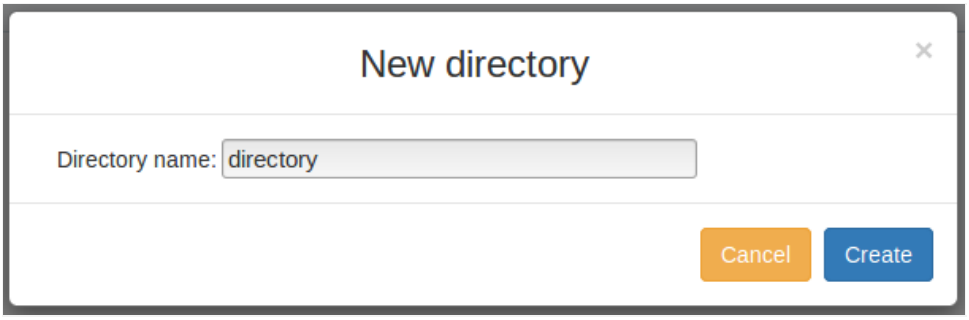
Clicking on “Editing”, you can create or upload files to the libraries. You can create files in a folder, with the goal to organize your files.

On Footer of the page, you can view the options to create or upload the files.



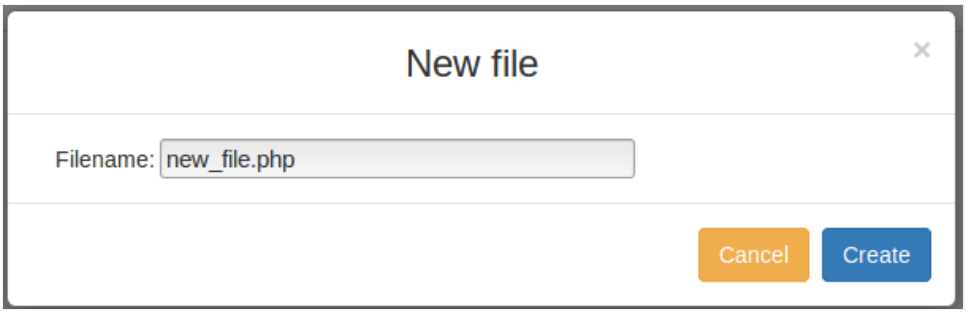
New Directory

To create a directory, click on “New Directory” and inform the name of the directory.



New File

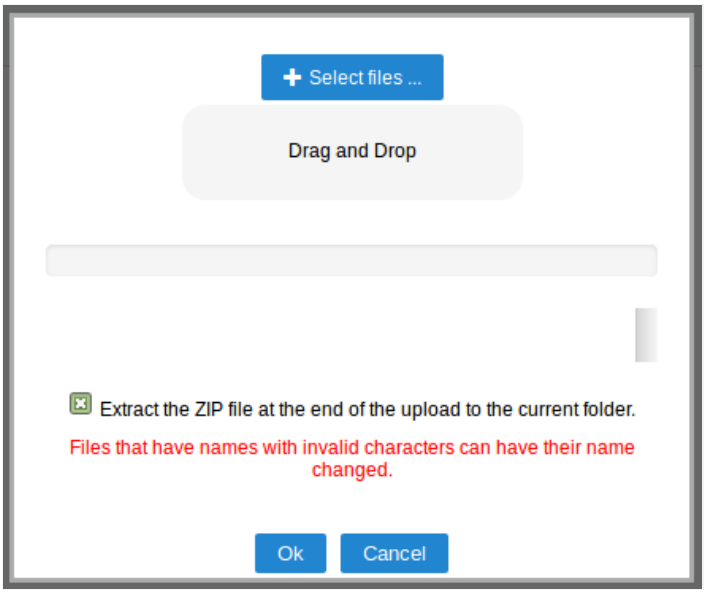
To create a file, click on “New File” and inform the name and extension (it is indispensable to inform the file extension).



Upload Files

This tool allows to upload files and external libraries and used them in the project.

Clicking on upload, select the desired file.



Delete

Delete the libraries and all the files contained in them.

Before deleting the library, be sure that it is not being used, you will not be able to recover deleted libraries.

Importing libraries into the applications

Like mentioned before, to use the libraries, you need to use the macro **sc_url_library()** and the **sc_include_library()** (Only for PHP files).

The macro **sc_url_library()** returns the path to the file in the library and the **sc_include_library()** is used to include php files into applications from the external library:

This example changes the background color of a grid application, if the code in the style.css file

```
.scGridPage
    {
        background-colo: #1c6f16 !important;
    }
```

Internal Libraries

This tool allows to create or upload PHP routines that can be reused in various applications of the project or other projects, depending on the level of access that they were saved.

There are three types of access levels that can be used.

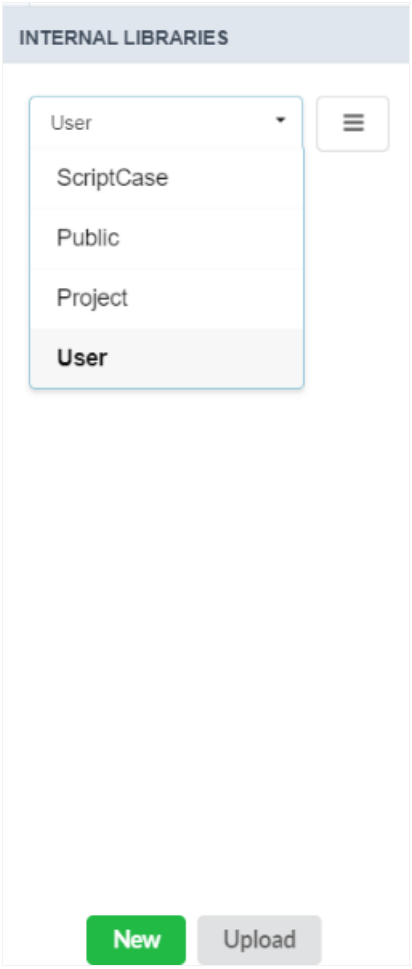
- **User** - Only for the user that created the script will have access.
- **Project** - All the users linked to the project will have access to the script.
- **Public** - All the users from ScriptCase will have access to the script.

You can't create or modify scriptcase that are in the Scriptcase Level, the scripts from this level are available to be used but nor modified.

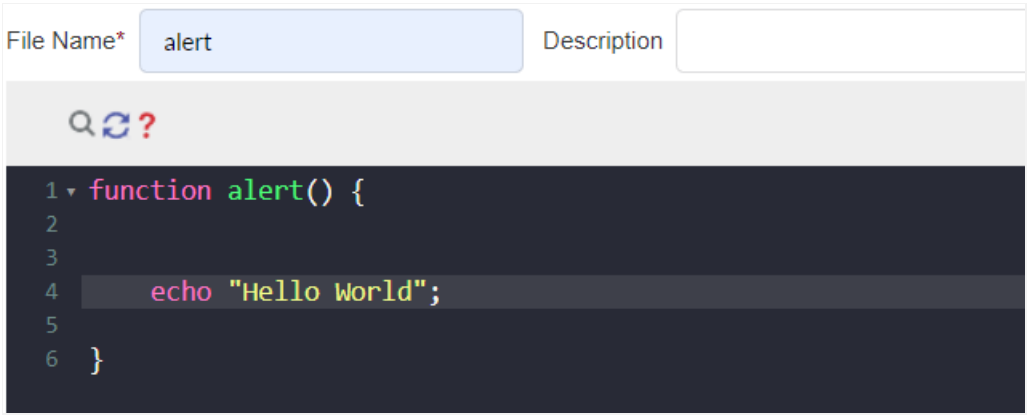
Creating a Script

The item **Internal Libraries** are accessed from the tools menu.

Select the level that the script is being saved, in this example the user level was used.

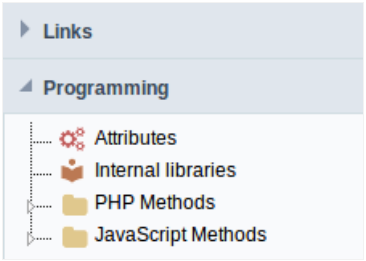


Next, inform a name for the file, place your code and save it.



Using the Script

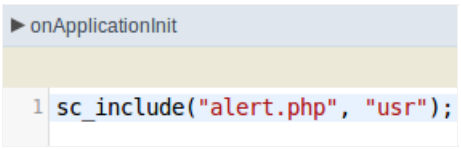
To use the Scripts, firstly you need to enable them in the applications that you want to use.



All the libraries that you have access will be listed, just select the the ones you desire to enable and use them.

To use the libraries, you need to import them by using the macro **sc_include**.

Example of using the macro: `sc_include("File", "Origin");`



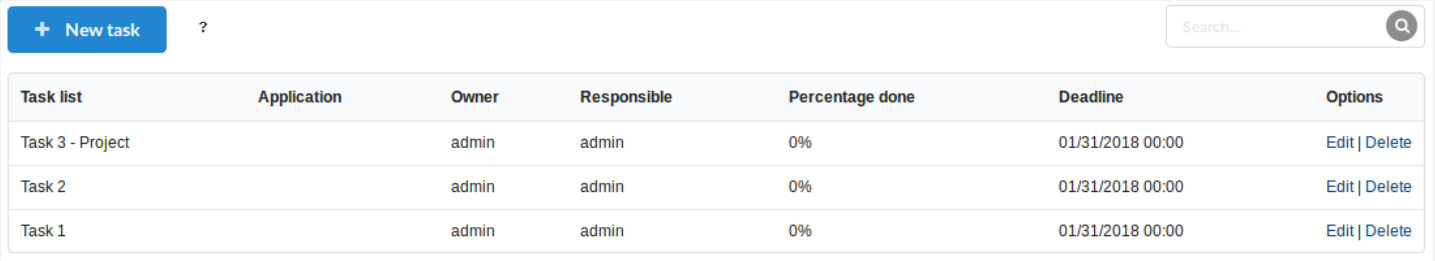
To-Do List

Tool that allows the creation of a list of tasks for the users registered in the same ScriptCase, with a progress control for the task.

A message that will always be displayed when the users that possess a task pending login.

To access the feature, go to the menu **Tools > To-Do List**

Accessing it, you will see all the registered tasks with the information about each one of them.

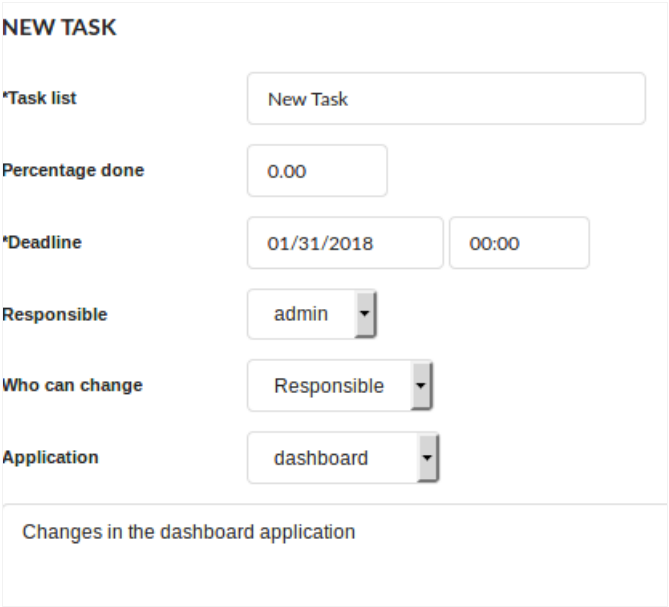


Creating a Task

The tasks can be created in four levels.

- **Public List** - All the ScriptCase users can view the tasks of the this level.
- **My List** - Only the Creator and the user responsible for the task can view.
- **Project: Public List** - All the users of ScriptCase, linked to the project that the task was created can view them. (This option is available only when a project is opened)
- **Project: My List** - Only the Creator and the Responsible user can view this task, they need to have access to the project where the task was created. (This option is available only when a project is opened)

Firstly, you need to select where you are going to create the task. Selecting the folder, the “New Task” button will be enabled.



Next, after clicking on “New Task”, you need to fill up the required fields.

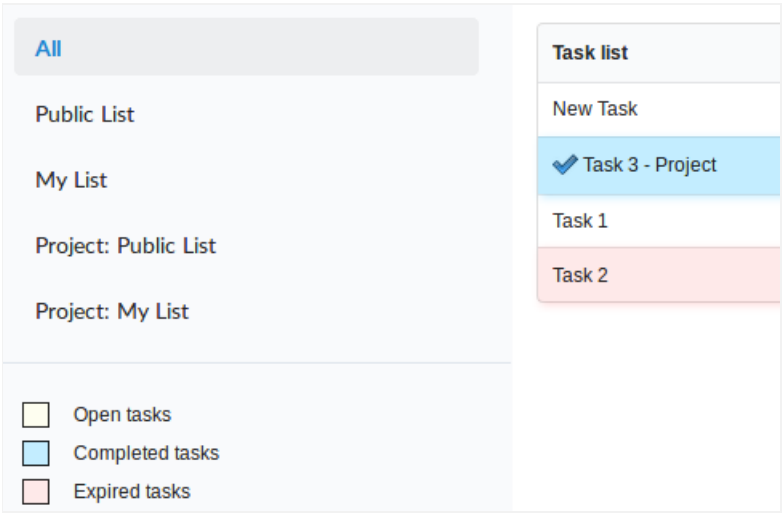
- **Task list** - Name of the task, which will be displayed in the Task list.
- **Percentage done** - Inform the percentage of the task conclusion.

- **Deadline** - Define the date and time that the task need to be concluded.
- **Responsible** - Define which of the ScriptCase users are responsible for the task.
- **Who can change** - Defines who can modify the current task:
  - **Owner** - Only the user that created the task.
  - **Responsible** - Only the user responsible for the task.
  - **All** - All the ScriptCase users can modify the task.
- **Application** - Inform the application of the project that the responsible user needs to edit. (This option will only be available for task created in a project.)
- **Description** - Task Description.

Task Status

This tool is used to inform the status of the task, using colors to symbolize the progress.

- Open tasks - Tasks recently created within the deadline, are represented in yellow.
- Completed tasks - Tasks that finished completely, are represented in blue.
- Expired tasks - Tasks that passed the deadline, are represented in red.



Messages

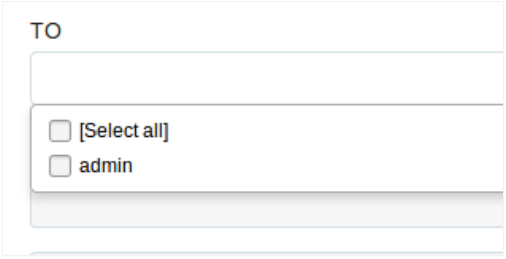
The message system allows the communication between the registered developers on the same ScriptCase.

You can access this feature. **Tools > Messages**.

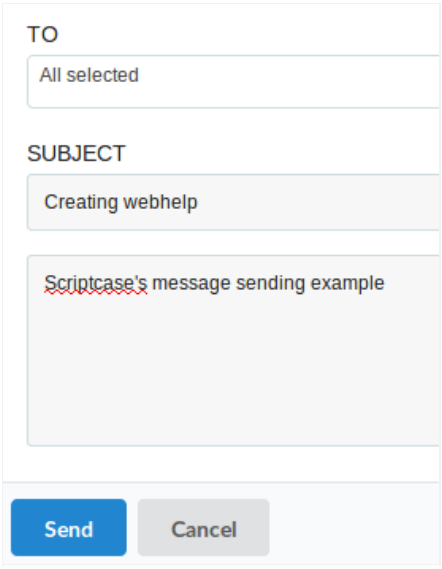
New Message

After clicking on the New Message button, you need to select the user that is going to receive the message.

One message can be sent to many users.

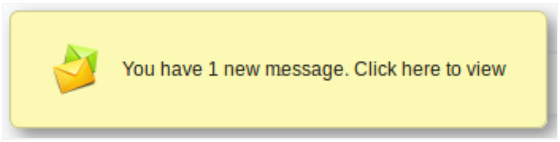


After typing the message, click on send.



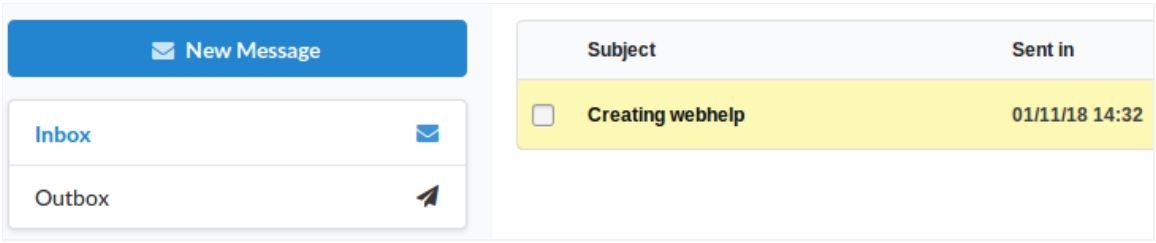
Inbox

Receiving the message, the users will be notified by an alert.



You can click on the notification bell or access **Tools > Message** to view the received messages.

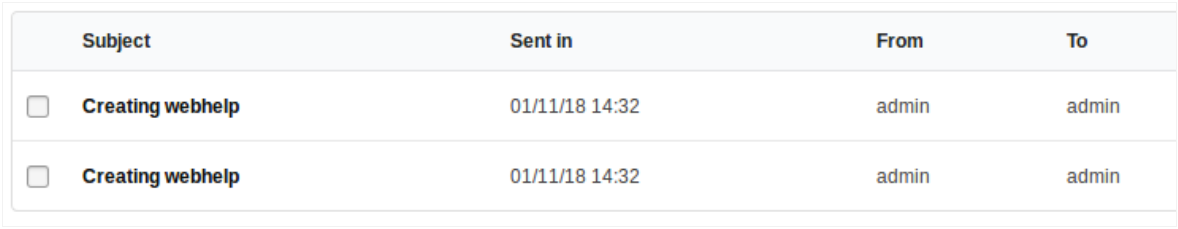
Accessing the inbox where all the received messages are stored and ordered by the date.



Outbox

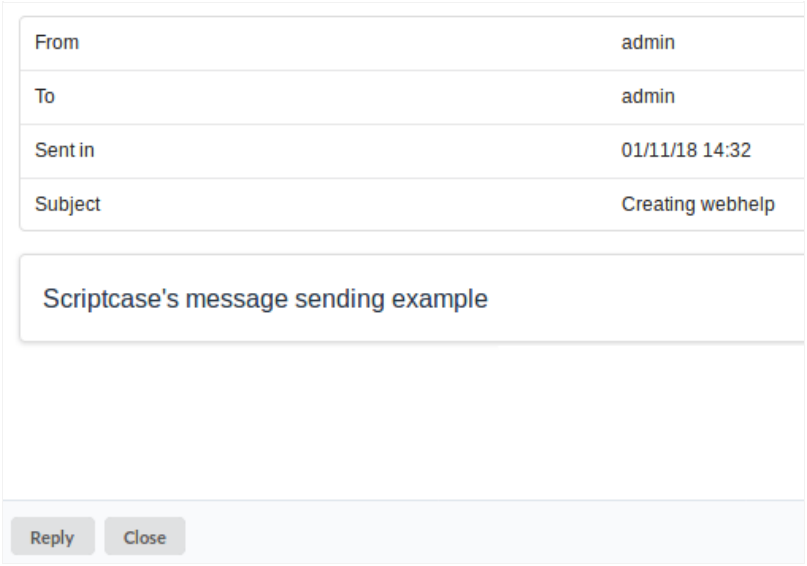
The sent messages are stored in the outbox.

By sending a message to various users, it will be registered a message for each user in the outbox.

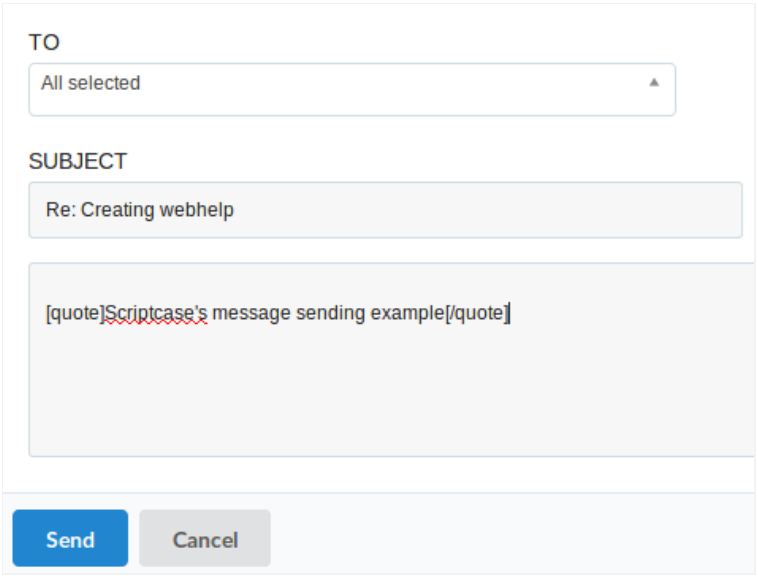


Read and Reply

Click on the message that you want to read.

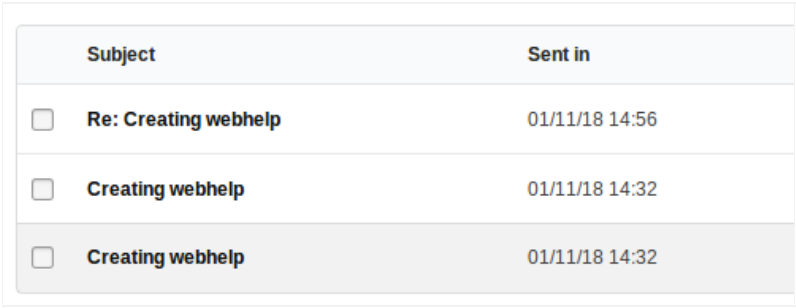


When reading a message, you can reply the received message.

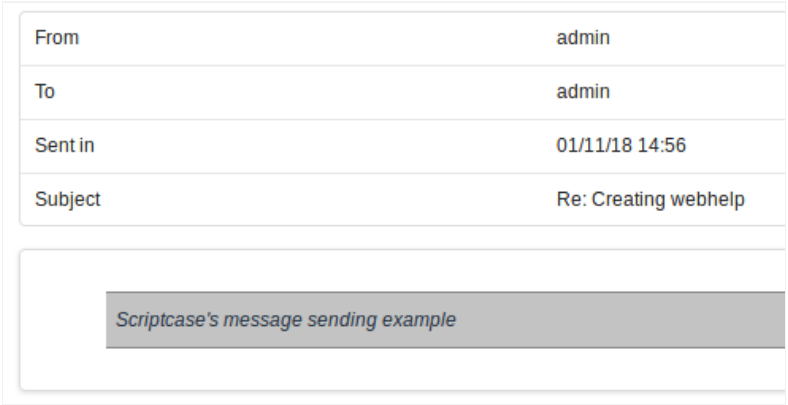


Received Response

In the Outbox, a responded message can be identified with a Re: before the subject

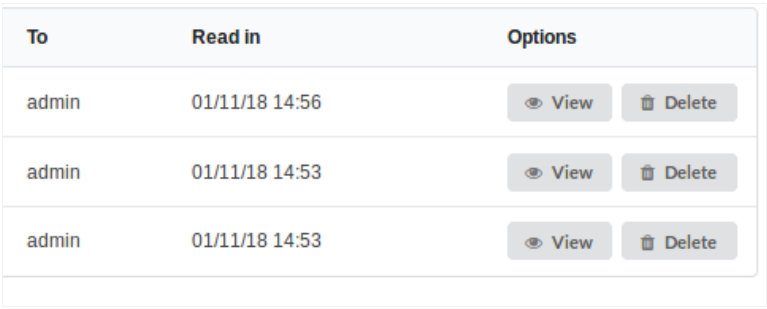


The original message, will be highlighted in the received response.

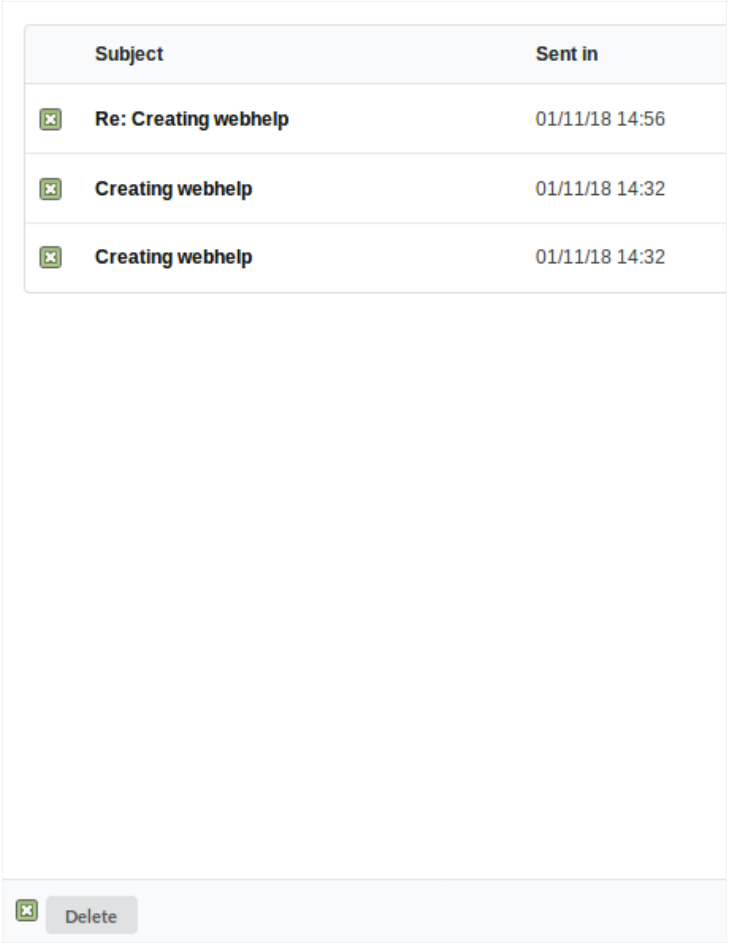


Delete Messages

You can delete the messages individually by clicking on the delete link.



You can also delete various messages. Select all the messages that you want to delete and then click on the delete button.



Converters

Project Converters

The Project Converter allows to convert the projects developed in previous versions of Scriptcase to the current version of Scriptcase.

Basically it is an import of the projects from different versions of Scriptcase.

The process of conversion does not modify any of the information of your project.

To prevent eventual problems, it is recommend to create a backup of your projects.

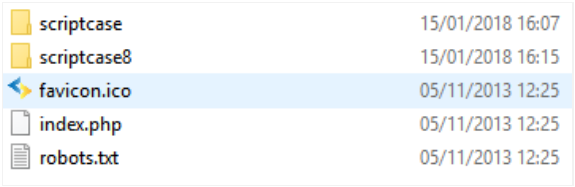
The project's conversion works the same way for all Scriptcase versions.

Coping Scriptcase

If your Scriptcase is not installed in the same place of your Scriptcase 9 you need to copy the Scriptcase folder, of the previous version, nexto to the current ScriptCase folder that will receive the projects.

In this example:

- **Scriptcase 8 folder** - Old version of the tool.
- **Scriptcase 9 folder** - Version that will receive all the projects.



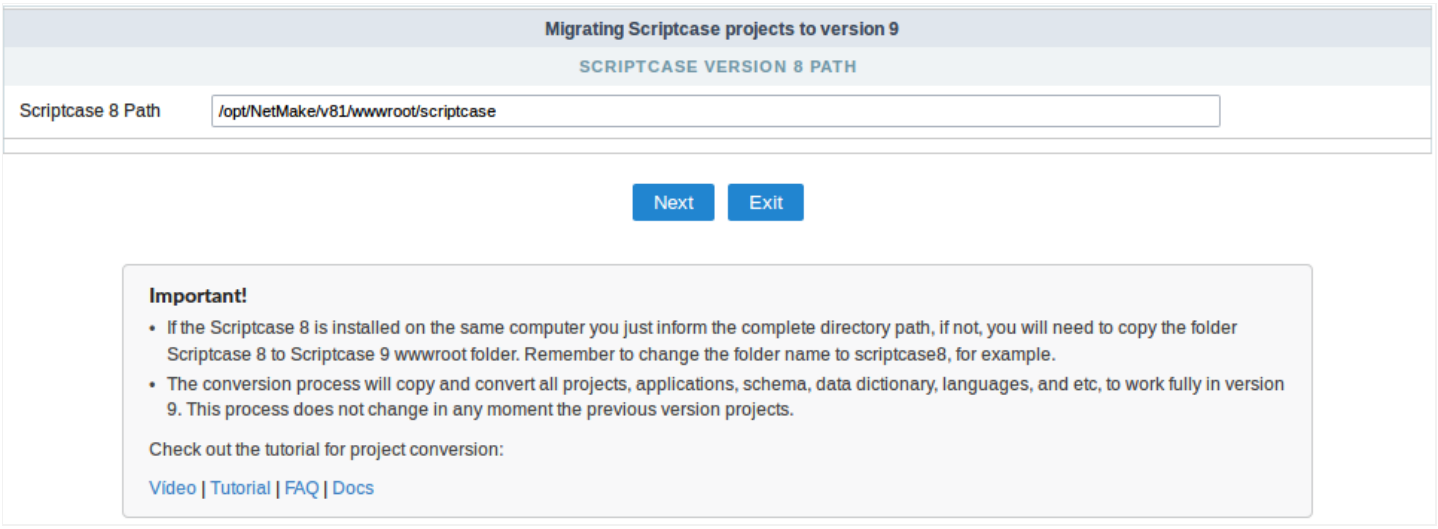
Coping the Scriptcase 8 folder to the same place

Path to Scriptcase

In Scriptcase select the converter (compatible to the version that you want to convert), and inform the complete path to the folder of the old version.

If your Scriptcase 9 is installed in the same place of your old version you can inform the complete path where your old version is installed.

- **Ex1: /opt/NetMake/v9/wwwroot/scriptcase**
- **Ex2: C:\Program Files(x86)\NetMake\v81\wwwroot\scriptcase**



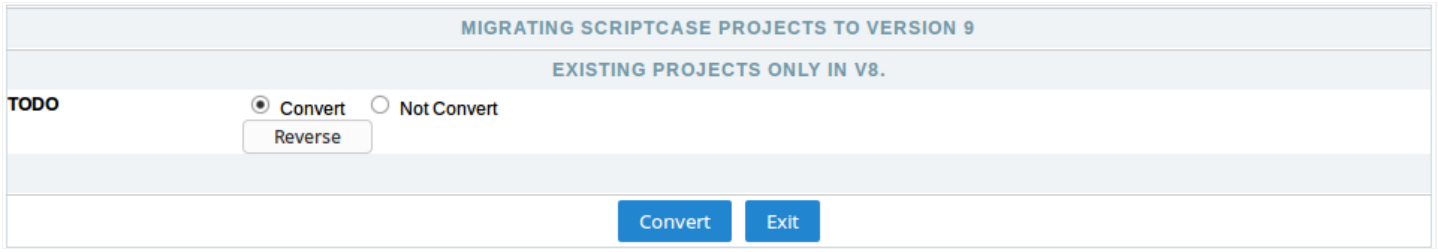
Informing the path to the Scriptcase 8

Converting the projects from your old version

In this screen will appear all projects from your old version.

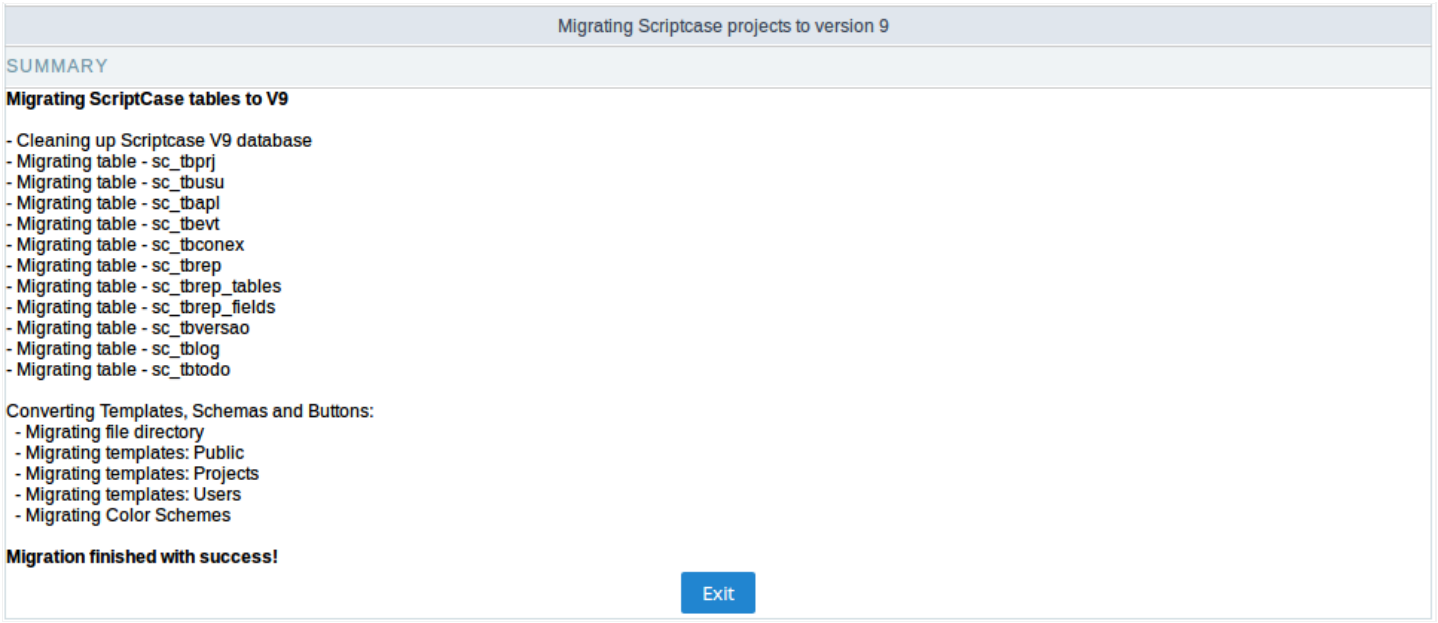
- **Convert**: Will convert the projects from your old version.
- **Not Convert**: Will not convert the projects from your old version.

If is not the first time that you are converting, may appear a option called **Overwrite**. If you convert with this option selected will overwrite your project with the old one.



Selecting the projects that will be converted

Next, you will view a log of everything that has been converted. With this the selected projects are already in the current ScriptCase and ready for use.

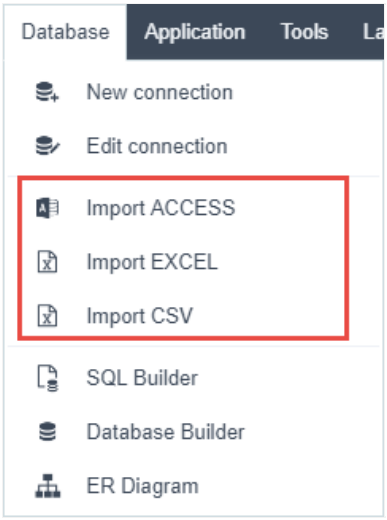


Information showing the success

Database Convert

Allows you to convert one or more data sources to one structured database. You can convert the stored tables in .XLS, .CSV, .accdb and .mdb files to the following databases: **MySQL**, **PostgreSQL**, **SQLite** and **SQL Server**.

You can find all the options to import databases in the menu **Database**.



Access Import

Selecting Files

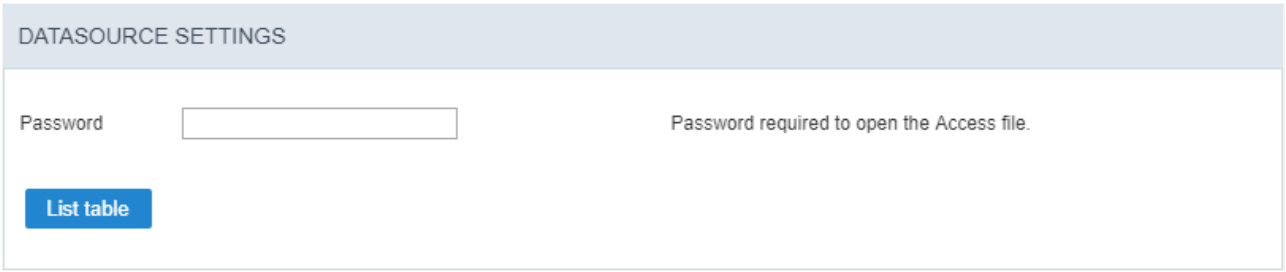
First, select the files that will be converted. You can inform the absolute path to the file, drag and drop the file or select the file in the option **Select Files**.

!mSelecting an access file to convert the tables

Configuring the tables

After selecting the file, you will need to access the configuration page.

On this page, you will see the password field, that is only used with **Acess** files that were selected and require a password to list the tables. In these cases, after informing the password, click on **list tables**.



Still in the same page, you can view the tabs **Tables** and **Advanced**.

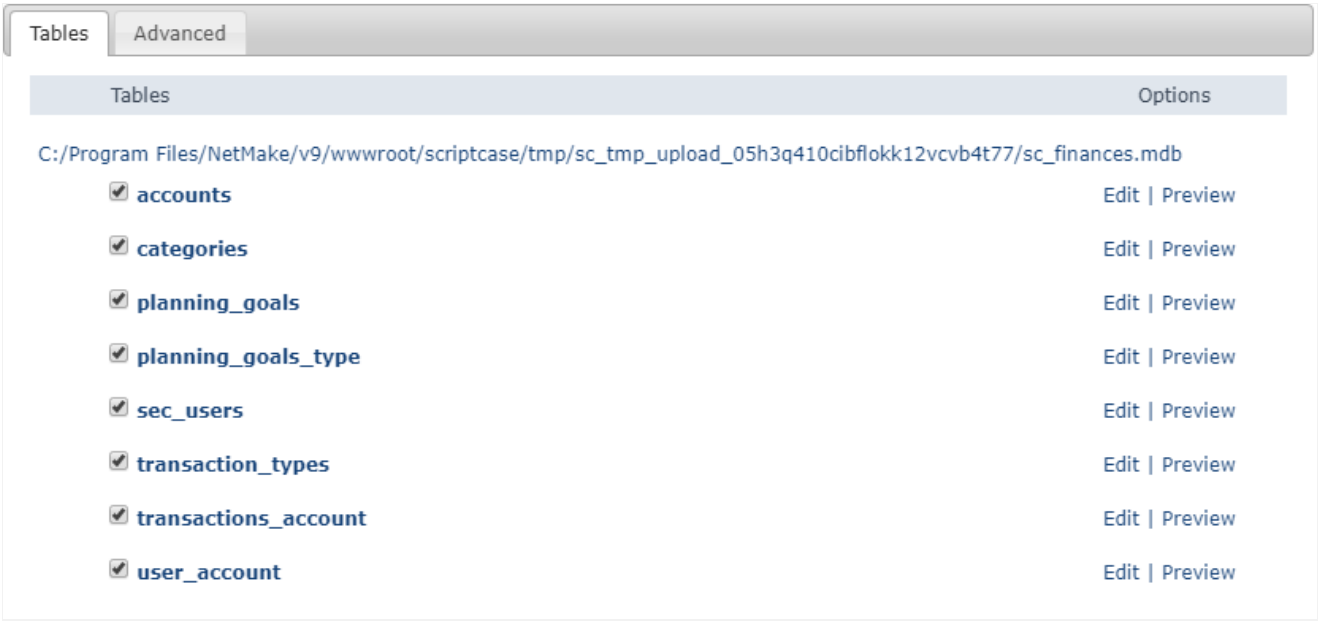
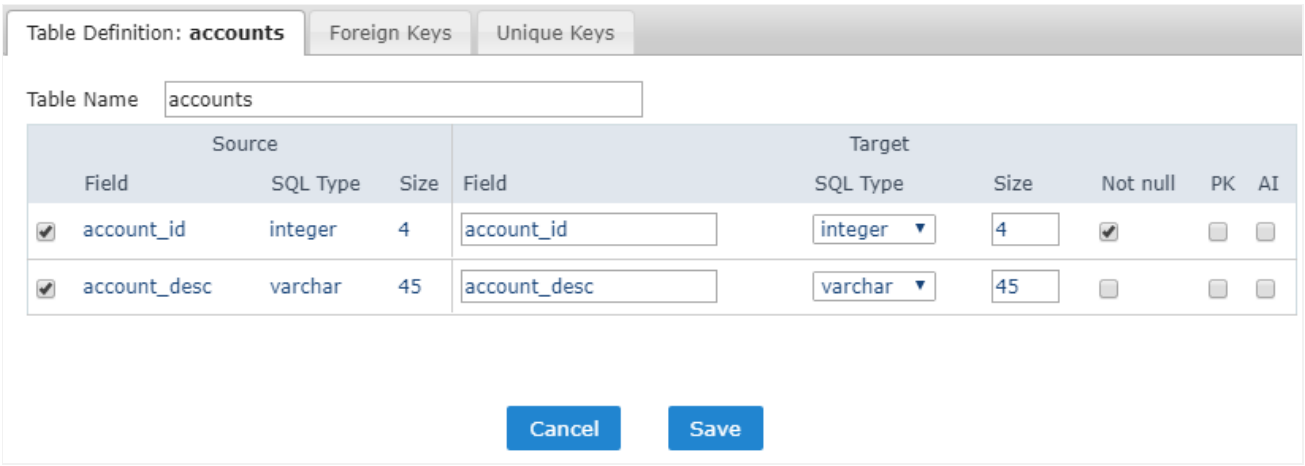


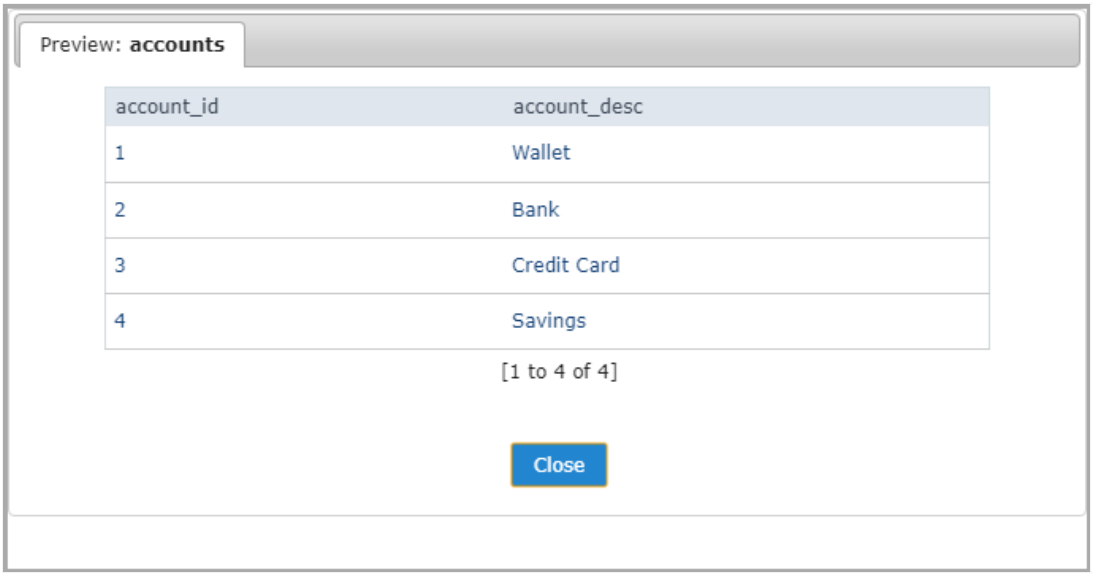
Table Tabs

We can view a list of the tables that can be converted. Beside the name of the tables, are two options: Edit and Preview.

On the option **Edit**, you can modify the table structure that will be created, like the name, field size, and the data type and other options.

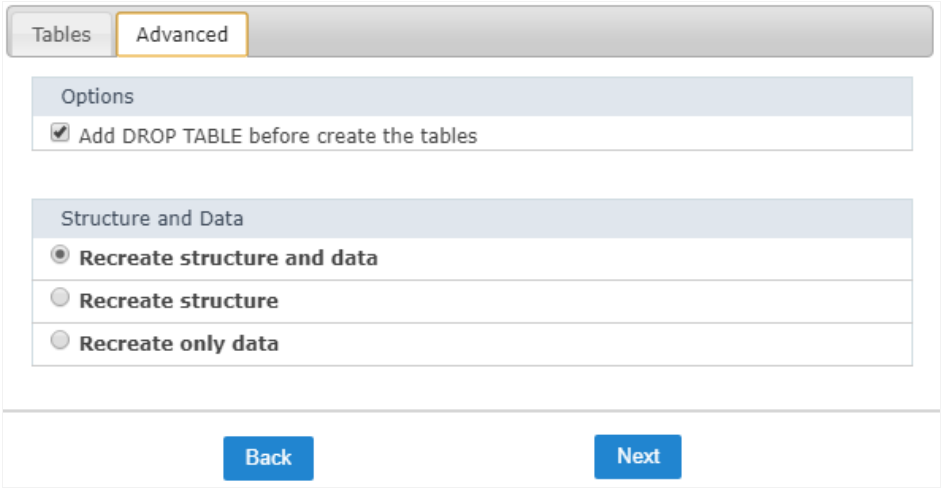


in **Preview** you will see a preview of how the table will be imported.



Advanced Tab

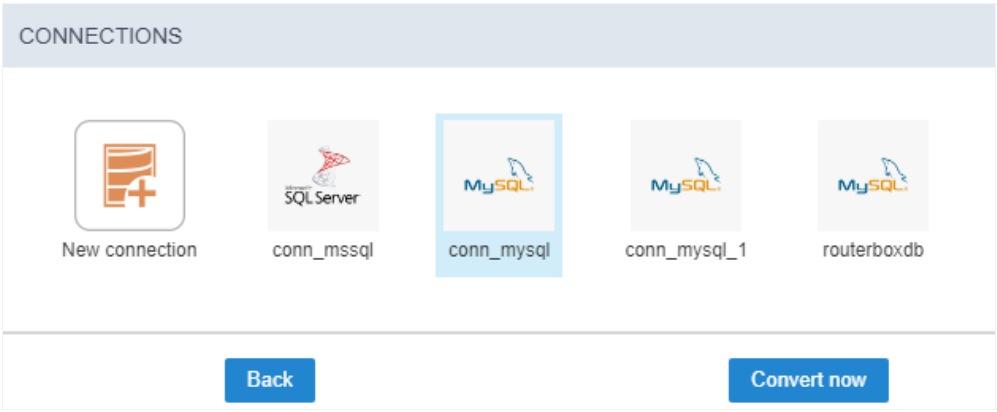
On the “Advanced” tab the configurations import script for the table are available. You can configure the script that will be created for the generation of the tables of new database.



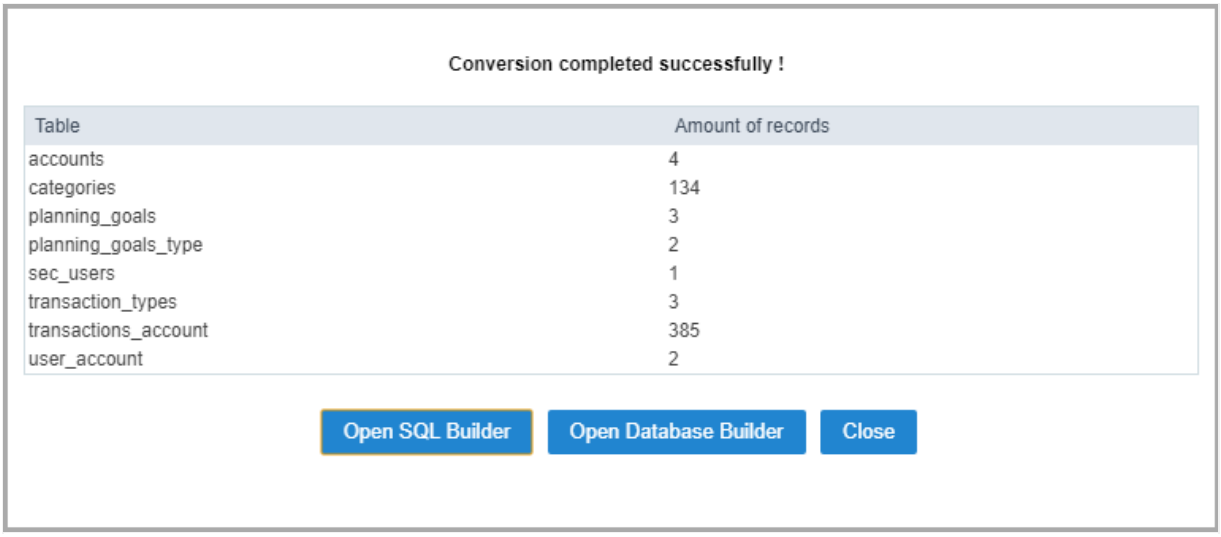
- Add DROP TABLE - Adds a DROP TABLE IF EXISTS before the CREATE of the tables, so that if the tables exist with the same name in the database, they will be deleted first.
- Recreate the database structure - Where you will define the table structure and the data that will be imported.
- Recreate the structure - Where you will define only the structure the table that will be imported.
- Recreate data - Where you will define only the data of the tables that will be imported (In case the tables exist in the database with the same structure of the ones being imported)

Connection

On this step, we need to select the connection (Database) to where you will import the tables. You can select a connection existing in the project or create a new one.



Next, you will be presented with a log of the tables that have been imported and the total of records inserted.



XLS Import

Selecting Files

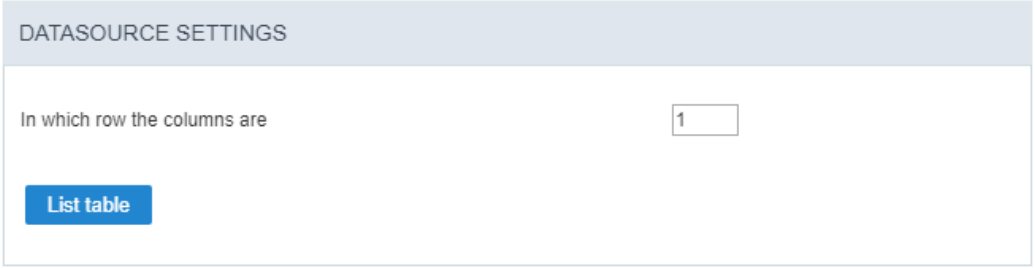
First, select the files that will be converted. You can inform the absolute path to the file, drag and drop the file or select the file in the option **Select Files**.

!mSelecting an access file to convert the tables

Configuring the tables

After selecting the file, you need to access the configuration page.

On this page, you need to inform first the line that the name of the columns are, this way all the information that is below will be considered a record of the table. If you do not inform the line, all the file content will be considered a record.



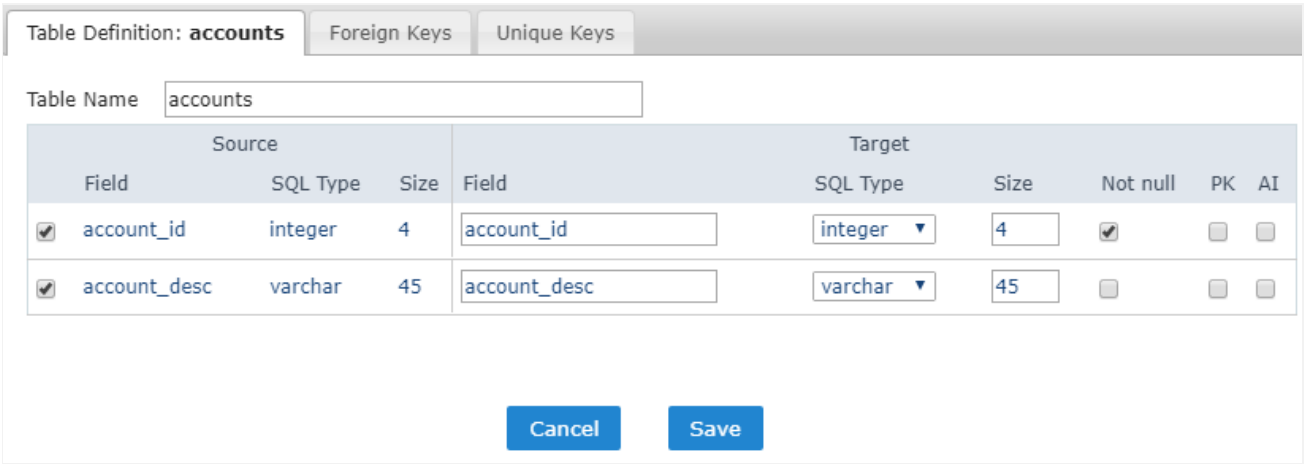
Still in the same page, you can view the tabs **Tables** and **Advanced**.

You can view below the list of tables of the files, with the names of the tables and the name of the file. In the case of XLS or CSV files, the table names are obtained through the name of the tab in the file.

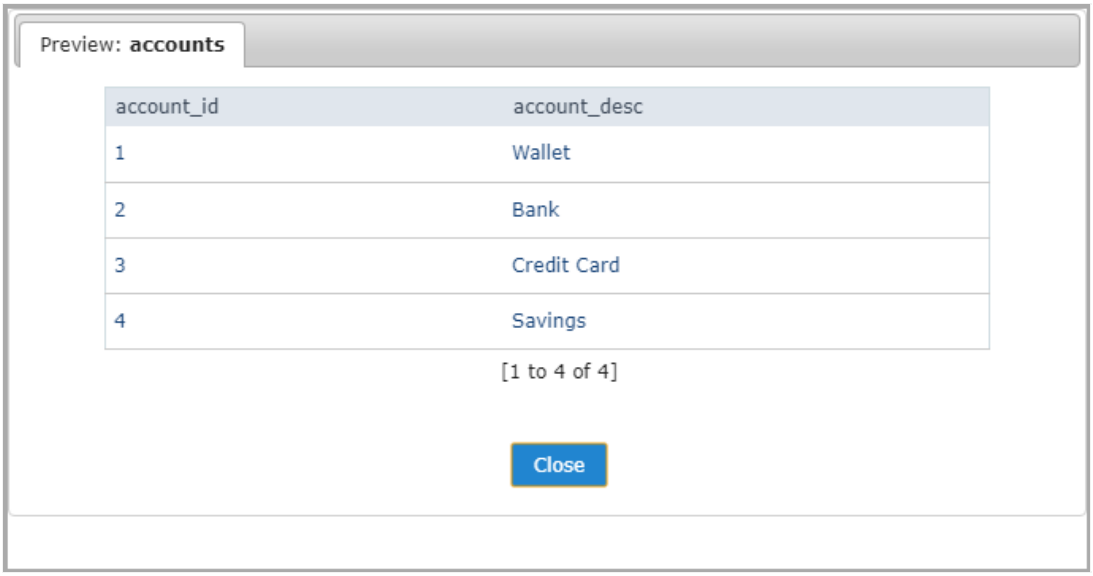
Table Tabs

We can view a list of the tables that can be converted. Beside the name of the tables, are two options: Edit and Preview.

On the option **Edit**, you can modify the table structure that will be created, like the name, field size, and the data type and other options.

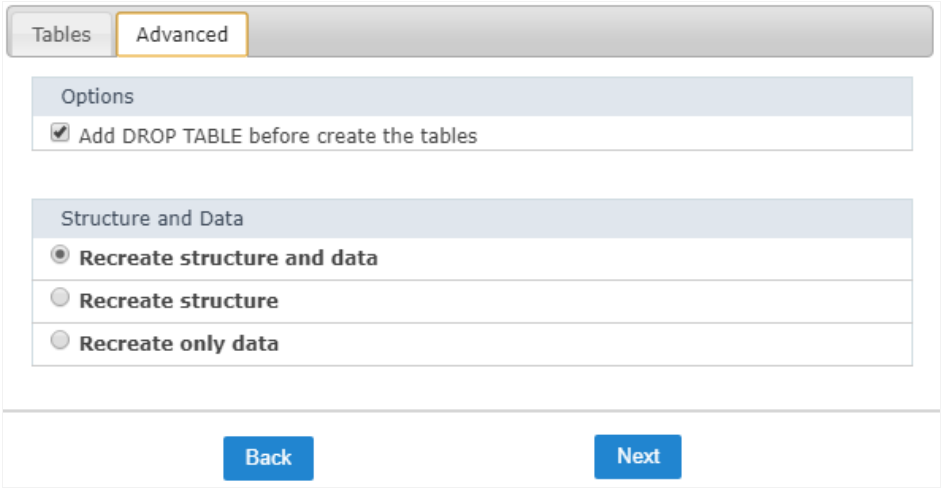


in **Preview** you will see a preview of how the table will be imported.



Advanced Tab

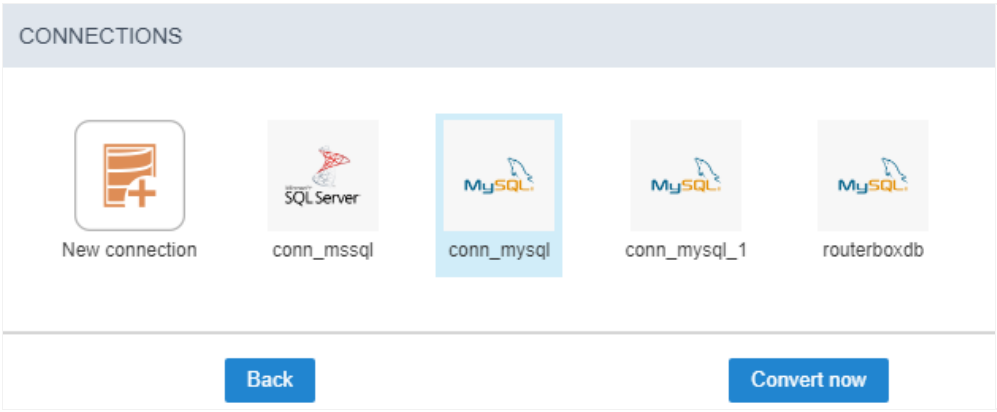
On the “Advanced” tab the configurations import script for the table are available. You can configure the script that will be created for the generation of the tables of new database.



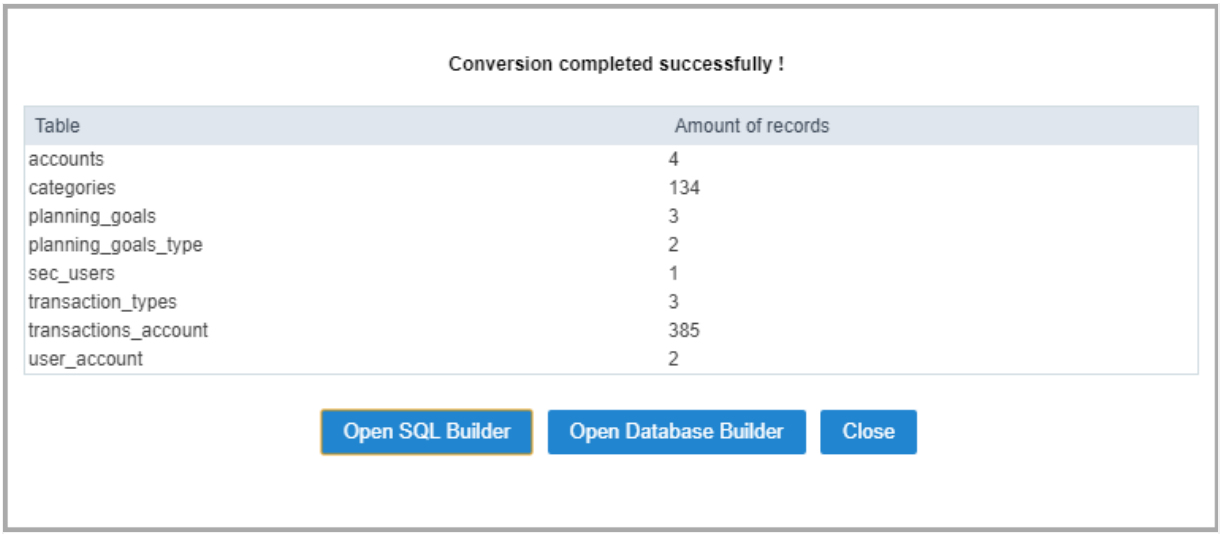
- Add DROP TABLE - Adds a DROP TABLE IF EXISTS before the CREATE of the tables, so that if the tables exist with the same name in the database, they will be deleted first.
- Recreate the database structure - Where you will define the table structure and the data that will be imported.
- Recreate the structure - Where you will define only the structure the table that will be imported.
- Recreate data - Where you will define only the data of the tables that will be imported (In case the tables exist in the database with the same structure of the ones being imported)

Connection

On this step, we need to select the connection (Database) to where you will import the tables. You can select a connection existing in the project or create a new one.



Next, you will be presented with a log of the tables that have been imported and the total of records inserted.



CSV Import

Selecting Files

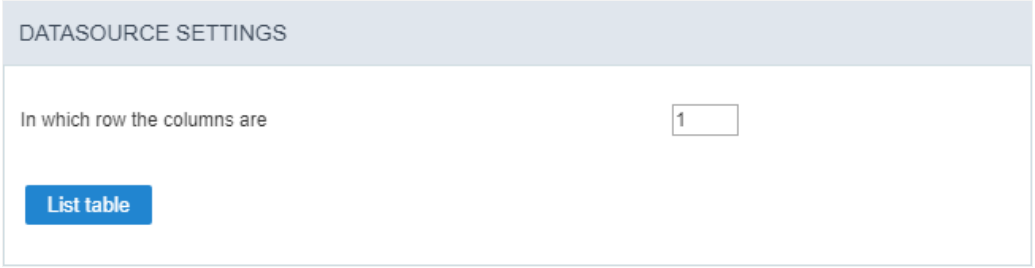
First, select the files that will be converted. You can inform the absolute path to the file, drag and drop the file or select the file in the option **Select Files**.

!mSelecting an access file to convert the tables

Configuring the tables

After selecting the file, you need to access the configuration page.

On this page, you need to inform first the line that the name of the columns are, this way all the information that is below will be considered a record of the table. If you do not inform the line, all the file content will be considered a record.



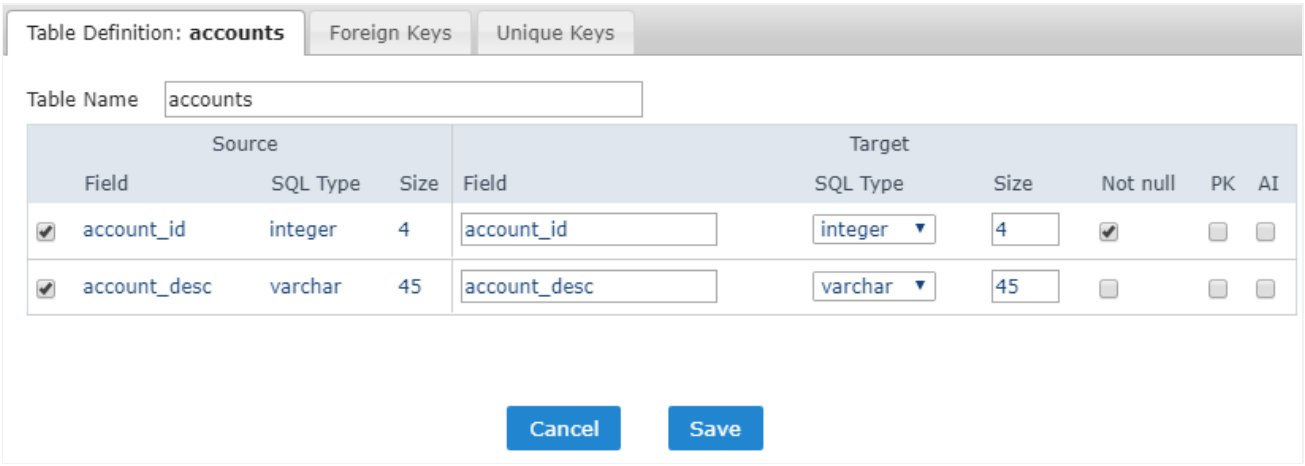
Still in the same page, you can view the tabs **Tables** and **Advanced**.

You can view below the list of tables of the files, with the names of the tables and the name of the file. In the case of XLS or CSV files, the table names are obtained through the name of the tab in the file.

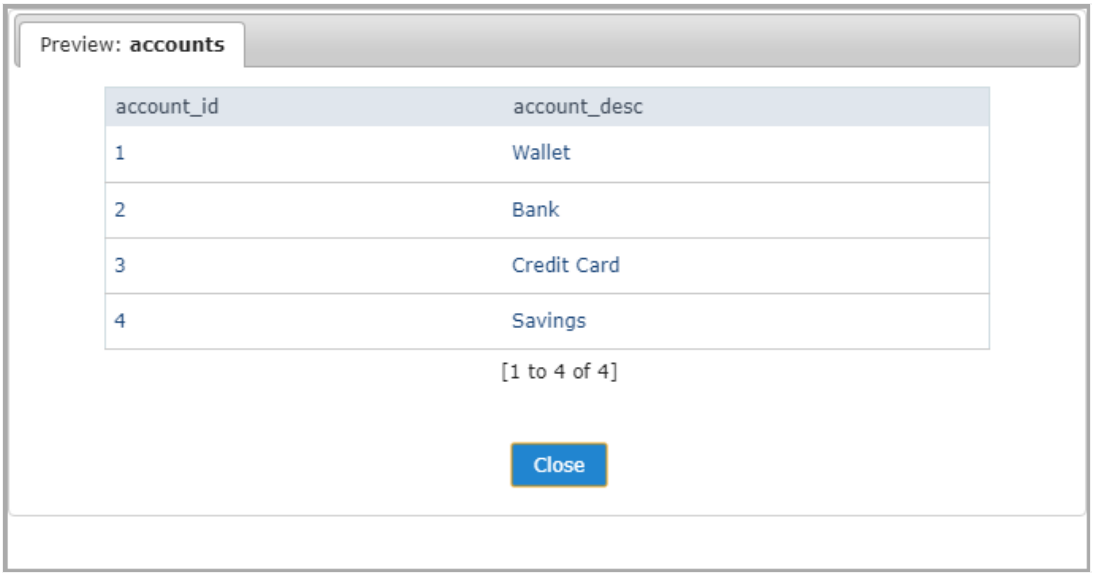
Table Tabs

We can view a list of the tables that can be converted. Beside the name of the tables, are two options: Edit and Preview.

On the option **Edit**, you can modify the table structure that will be created, like the name, field size, and the data type and other options.

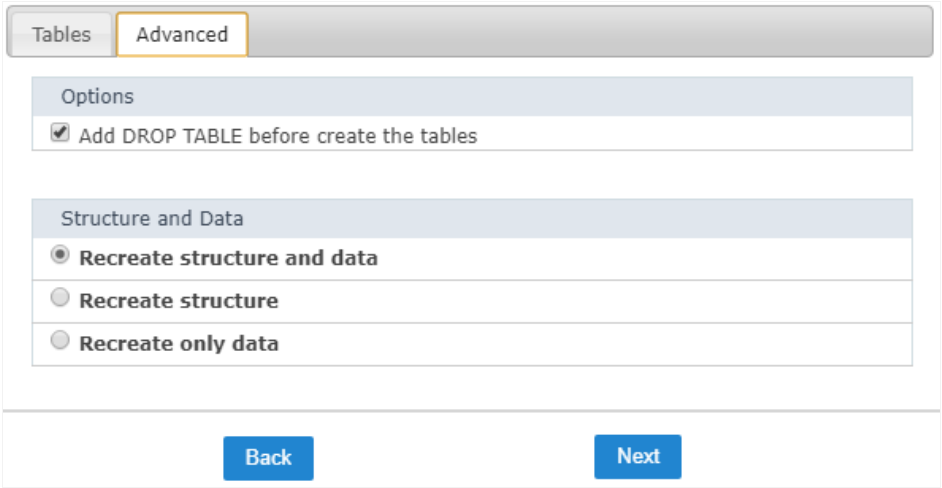


in **Preview** you will see a preview of how the table will be imported.



Advanced Tab

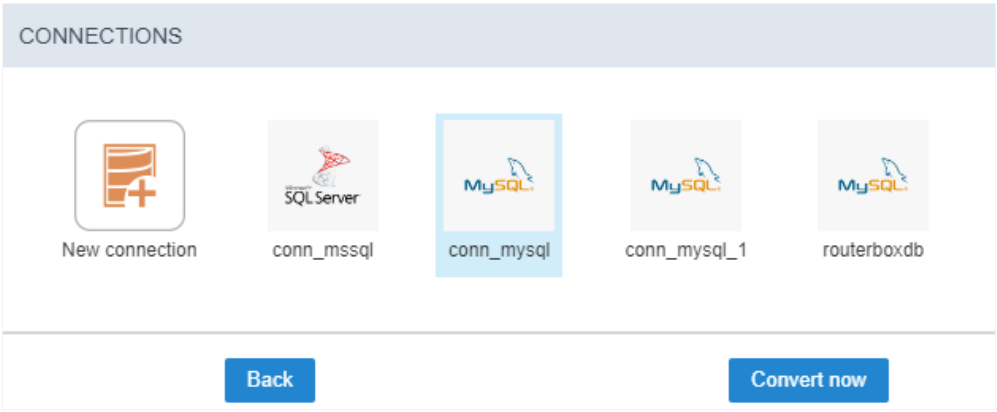
On the “Advanced” tab the configurations import script for the table are available. You can configure the script that will be created for the generation of the tables of new database.



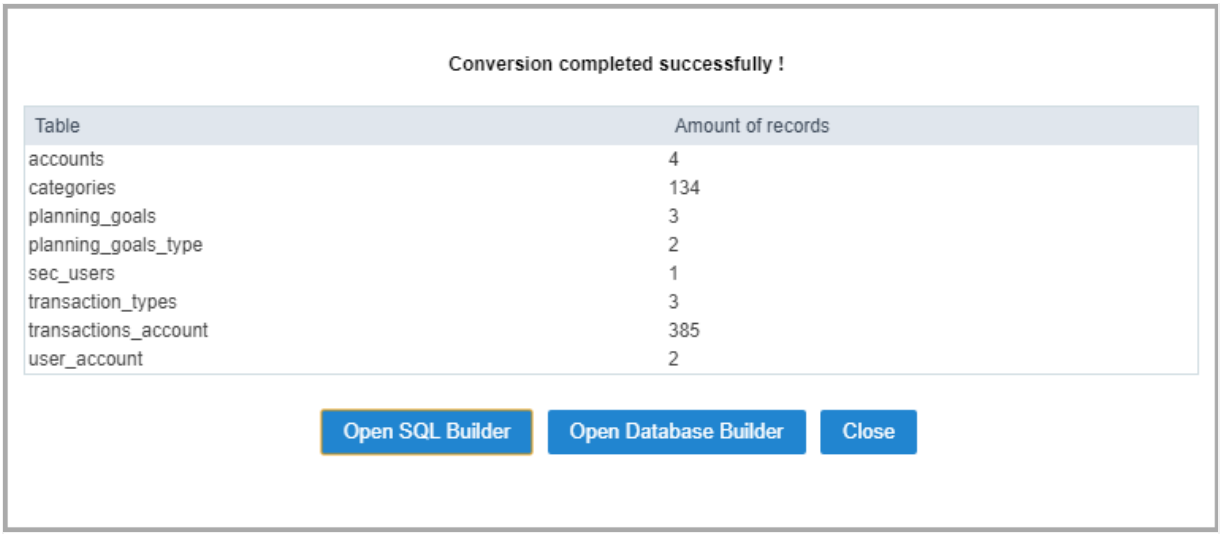
- Add DROP TABLE - Adds a DROP TABLE IF EXISTS before the CREATE of the tables, so that if the tables exist with the same name in the database, they will be deleted first.
- Recreate the database structure - Where you will define the table structure and the data that will be imported.
- Recreate the structure - Where you will define only the structure the table that will be imported.
- Recreate data - Where you will define only the data of the tables that will be imported (In case the tables exist in the database with the same structure of the ones being imported)

Connection

On this step, we need to select the connection (Database) to where you will import the tables. You can select a connection existing in the project or create a new one.



Next, you will be presented with a log of the tables that have been imported and the total of records inserted.

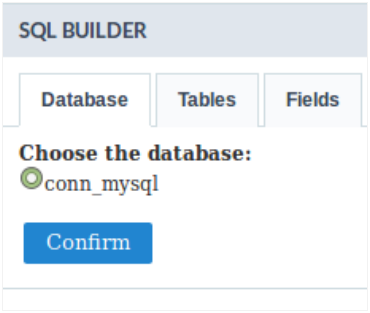


SQL Builder

This tool helps the construction of Queries in a Wizard form. These queries can be saved for later use. Any SQL can run directly on the Run tab that is displayed after the database is chosen.

Database

Firstly you need to select the connection(Database) that you want to use. **It is important that you must have a connection created in the project for it to be listed in SQL Builder.**



After selecting the desired connection, click the **Confirm** button to use this connection.

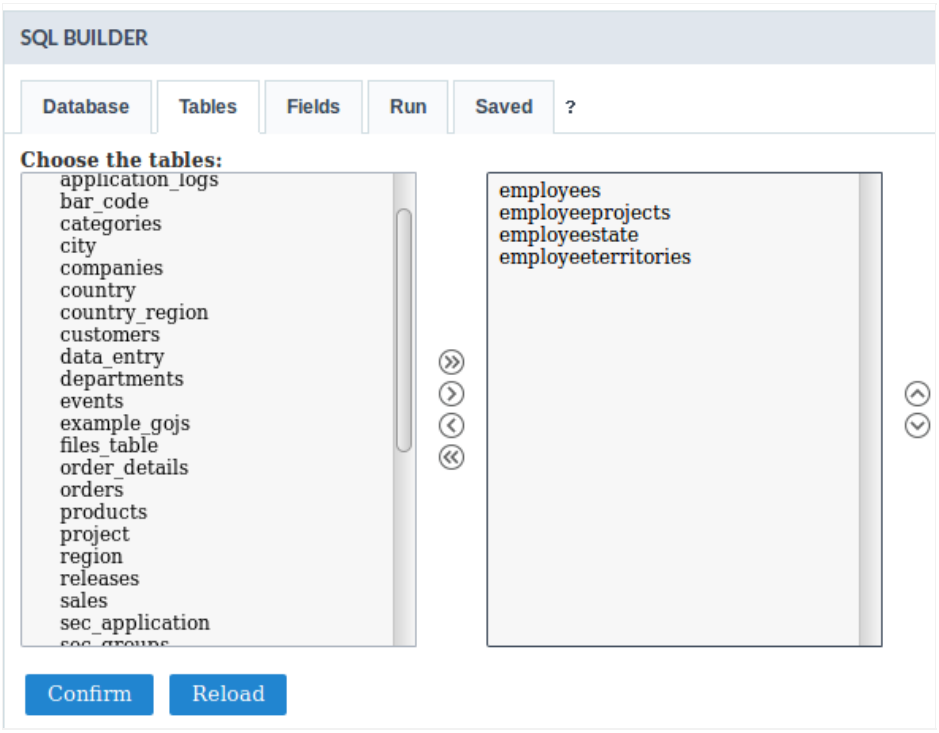
WARNING: The SQL Builder does not parse SQL, only receives the SQL Query informed by the user and executes it. So, if you use PDO drivers, only these are enabled: **PDO_DBLIB**, **PDO_MYSQL**, **PDO_PGSQL**, **PDO_SQLITE**. If you use a driver other than one of these listed, you may be returned with the "**BAD GETCOLUMNMETA()**" error in the column titles.
Go to

PHP page
for more information.

Tables

All the existing tables in the database are listed in the **Tables** tab. You can select various tables, although they need to have a relationship between them.

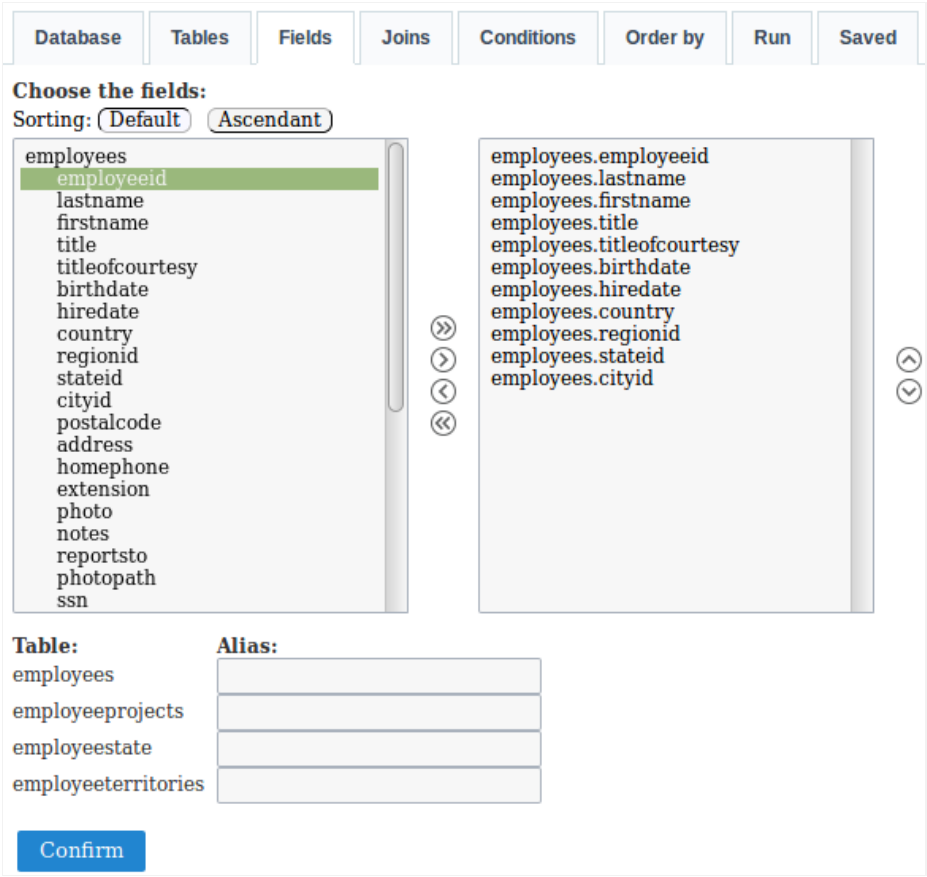
Select the desired tables to create the query.



After selecting the tables, click the **Confirm** button to use this connection.

Fields

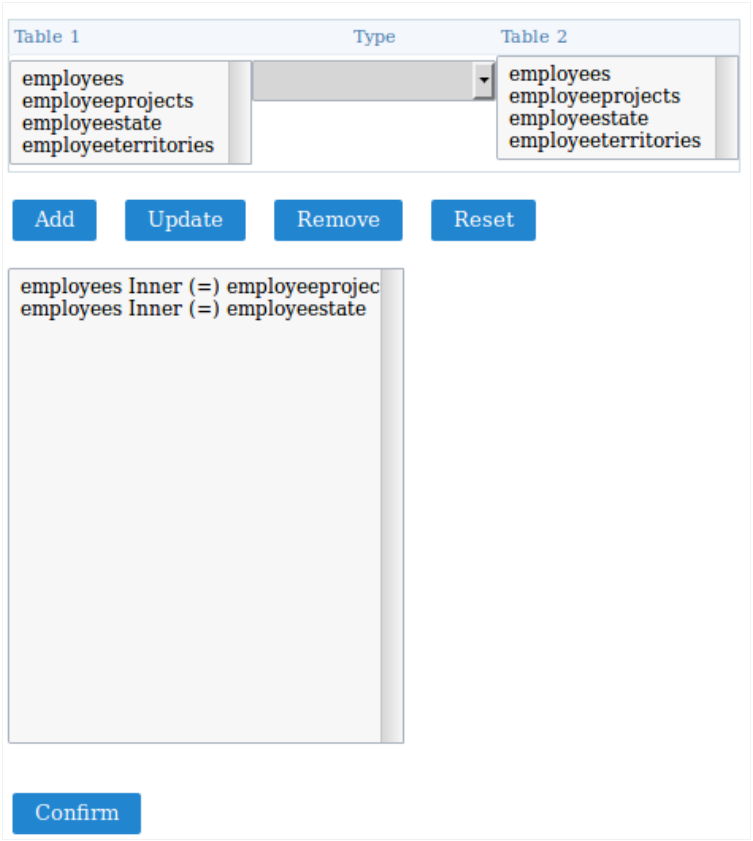
Select the fields that will be part of the query. The fields can be ordered alphabetically (Ascendant) or in the order that they were created (Default)



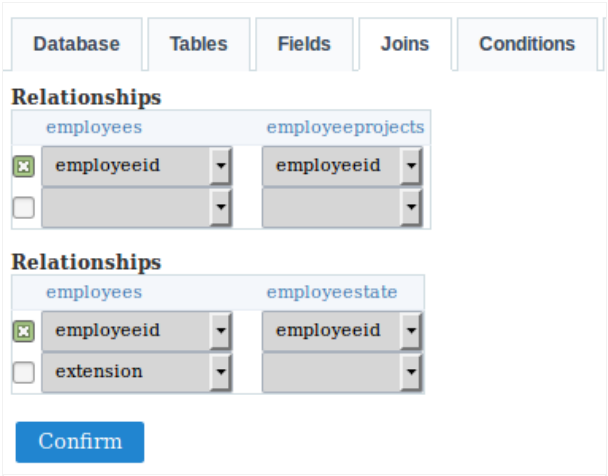
It is possible to create an **alias** for the selected tables, this option can be found below the field list were also all the selected tables will be listed.

Joins

Defining the fields, it is necessary to configure the join. To do that, you need to select the tables that have a relationship, next you will select the type of join that will be used in the field **Type** then click on Add.



After clicking on add, click on **Confirm**, so that you can inform the fields that do the relationship between the tables.

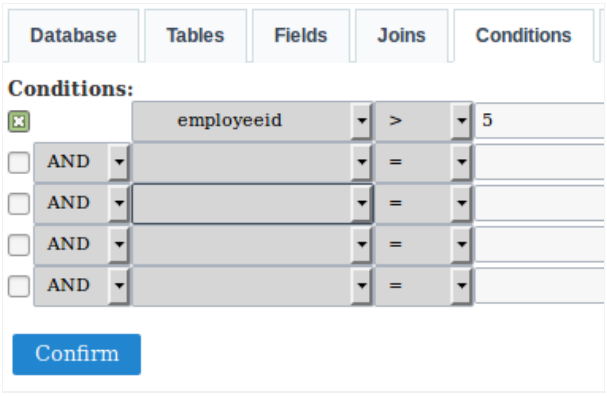


Defining the fields, you will be redirected to the run tab.

To add a “WHERE” clause to the SQL, click on the tab **Conditions**

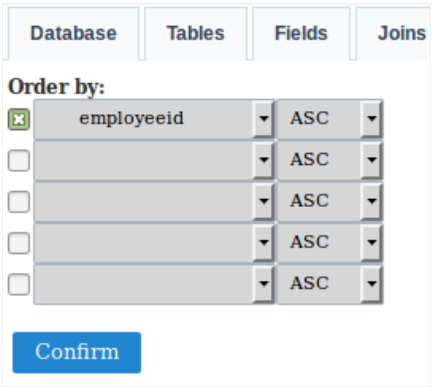
Conditions

You can add the “WHERE” clauses where the query was created.



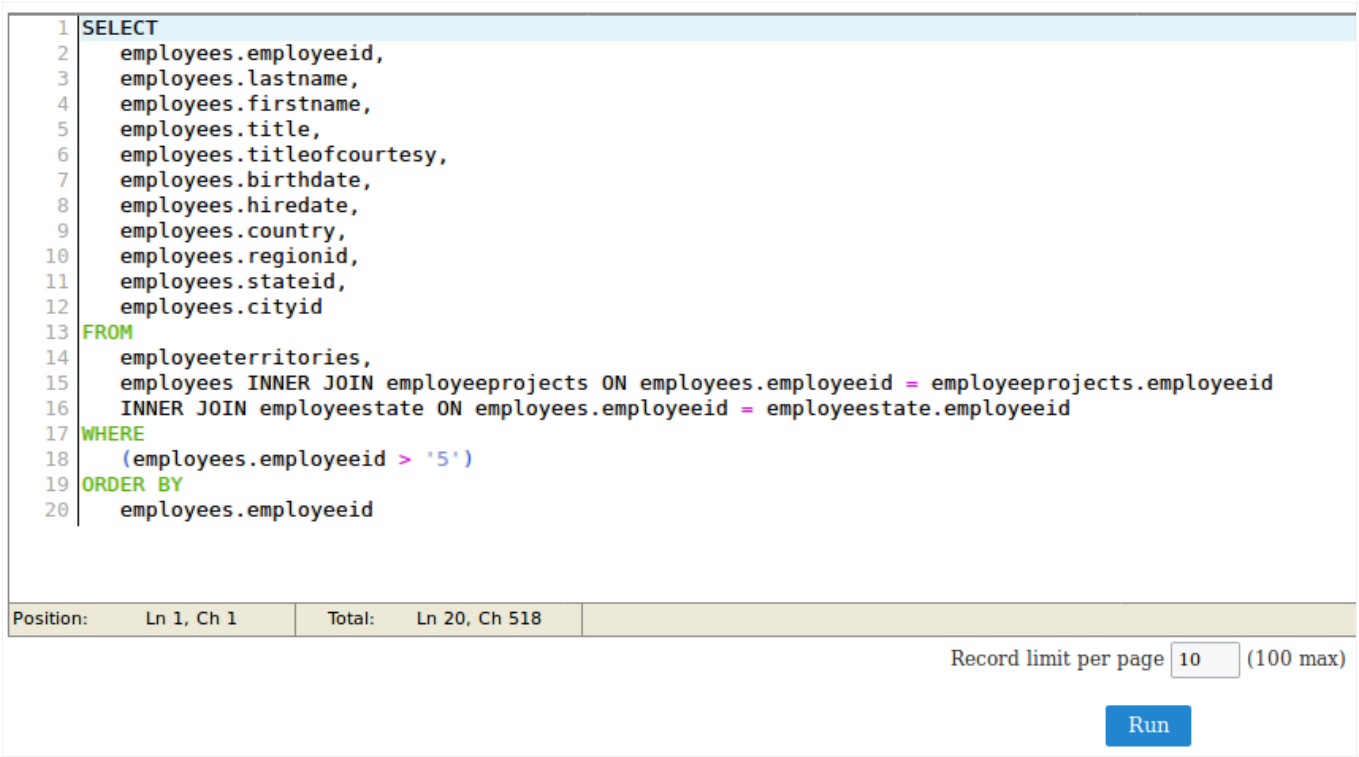
Order By

Define the sorting of the query result display.

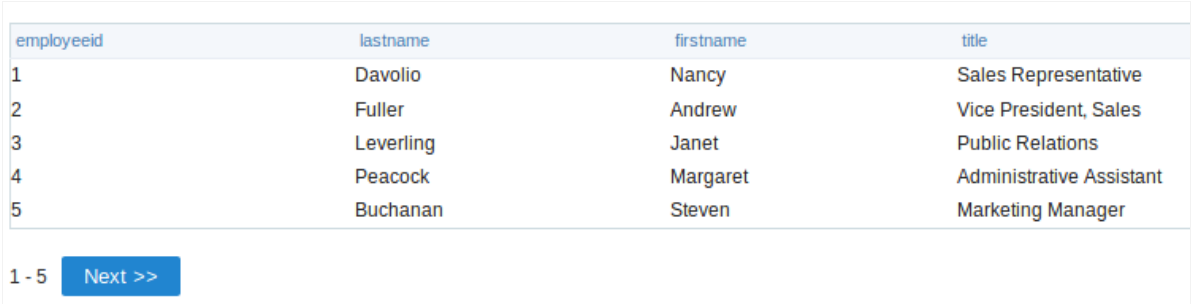


Run

This is where you can run the SQL commands. The result of the query configuration wizard will be displayed on this tab. You can run DML (Data Manipulation Language) commands here and also DDL (Definition Data Language) commands.



You can define a total of records per page.

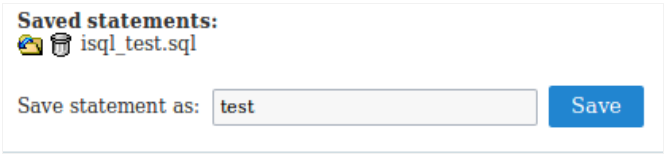


employeeid	lastname	firstname	title
1	Davolio	Nancy	Sales Representative
2	Fuller	Andrew	Vice President, Sales
3	Leverling	Janet	Public Relations
4	Peacock	Margaret	Administrative Assistant
5	Buchanan	Steven	Marketing Manager

1 - 5 Next >>

Saved

To save the created commands, just inform the name of the query and click on the save button. The commands that are already, will be listed.



Database Builder

This is a tool that allows the management of your database into ScriptCase. This tool is avaliable to connections with: MySQL, SQLServer(Cannot be ODBC), SQLite, Oracle and PostGreSQL.

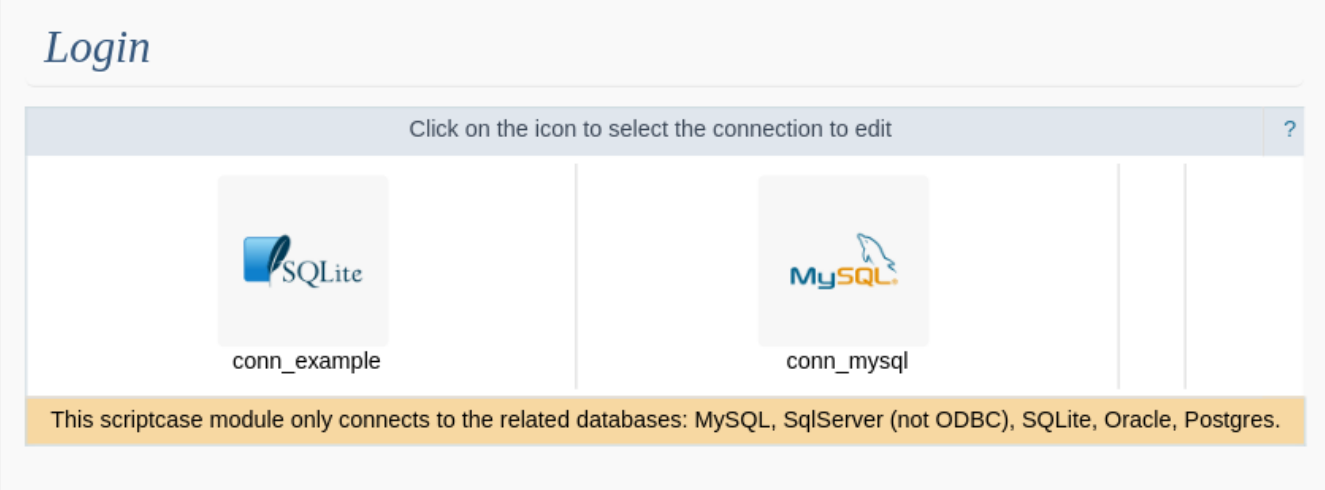
In the beginning we must select an existing connection in the project.

This tool can be accessed in the menu **Database**.

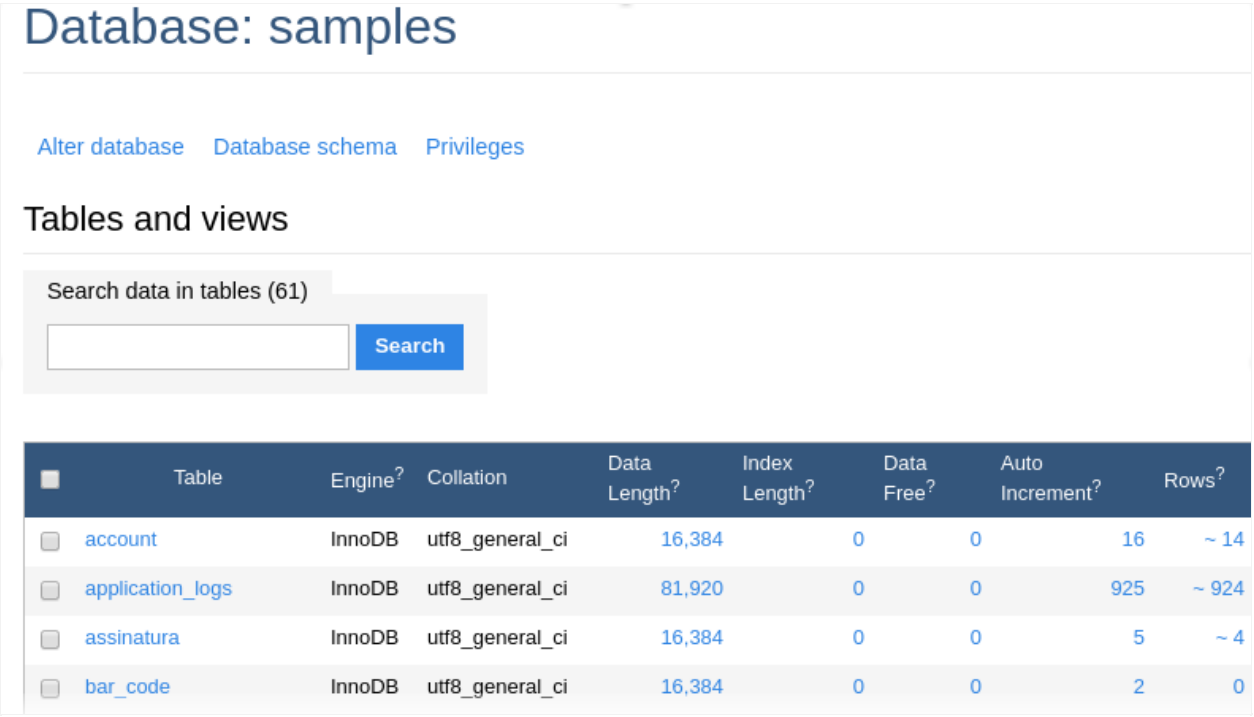
The options and settings on the Database Builder depends of the user database privileges and the used database.

Accessing the Database Builder

First, select the database connection that we wish to display.



Next step, we will go to the home screen from Database Builder, where we can access all the available functionalities.



At the screen left corner, we will see a few functionalities and the table list, to a quickly access of the selected database.

DB

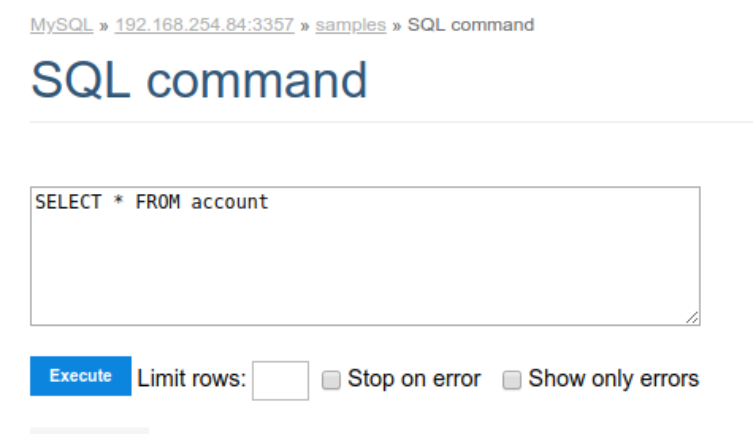
Allows see the database that is being accessed in the moment. Here we can change the database.

To the databases which possess user access restriction, it is necessary that the user set in the connection creation has access to all the databases

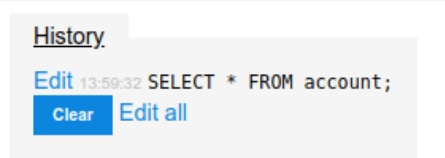
SQL Commands

Allows to execute any SQL command, from a select command to procedure commands of creation and execution, for example.

In this case, we will execute a simple SELECT.



Below the code insertion area, we can see a history of commands that were executed.



If everything goes well, when runs the command, will be showed the consult result, with the edição dos registros possibility.

```
SELECT * FROM account
```

accountid	accountdescription
1	OFFICE EQUIPMENT
2	FIXED TELEPHONY
3	MOBILE TELEPHONE
4	SOFTWARE
5	SERVICE ADVISOR
6	MAIL
7	CROSSING AIR
8	TAXI
9	TIKET MEAL
10	ELECTRICAL ENERGY
11	WATER
12	RENTAL
13	LODGING
14	HARDWARE

-1 rows (0.005 s) Edit, Explain, Export

Below of the displayed result, we will see a summary of the used tables in the SQL, informing the number of lines, the quantity of id fields and other informations, even the executed SQL.

17 rows (0.001 s) Edit, Explain, Export

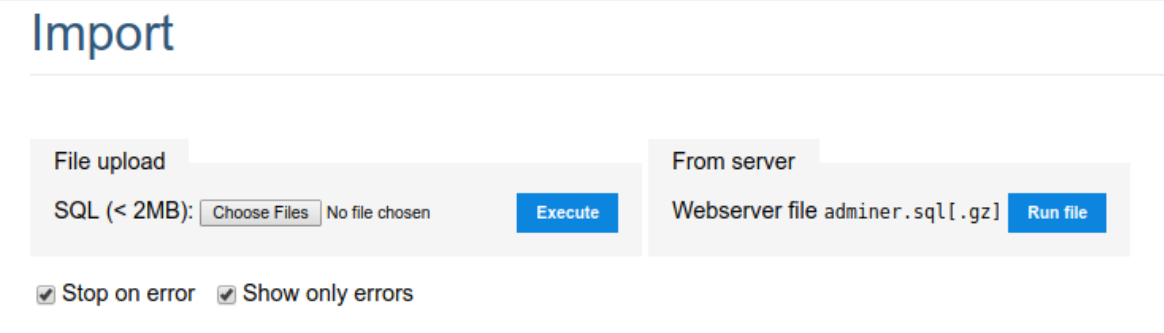
id?	select_type?	table?	partitions?	type?
1	SIMPLE	account	*NULL*	ALL

```
SELECT * FROM account
```

To return to the database builder home screen, select the table in the 'breadcrumb' in the page top.

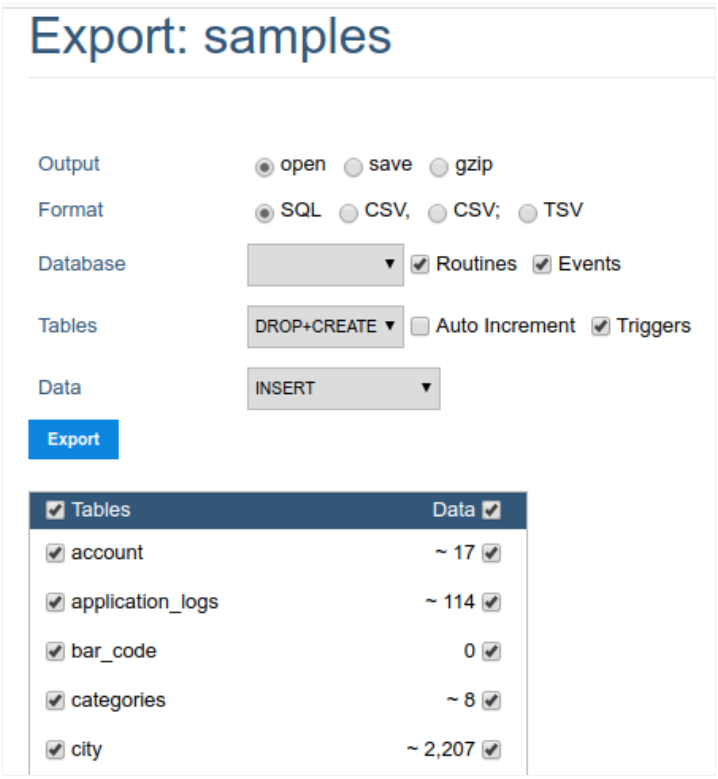
Import Files

Allows you to execute .SQL files.



Export Tables

Enables the creation of a dump in the database.

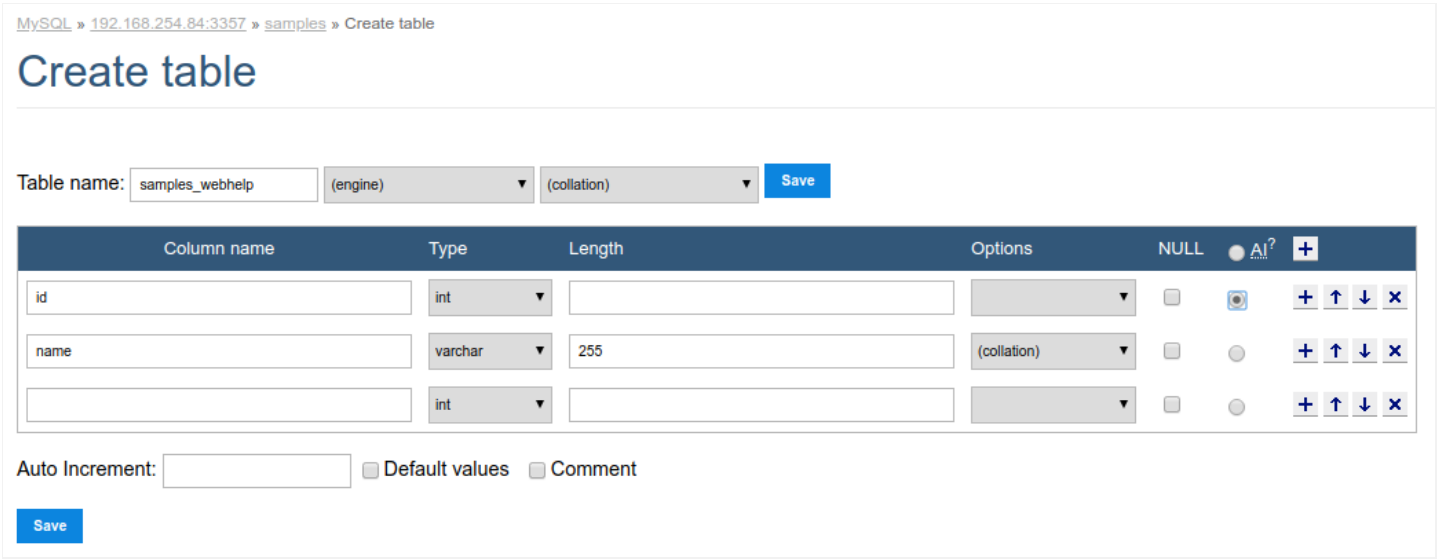


We must review the default settings of the exportation, for that we can change them and format according with our necessities.

- Output - Actions around the exportation.
  - Open - Display all the generated SQL inside the ScriptCase.
  - Save - Generates a `.sql` file.
  - gzip - Generates a `.sql.gz` file.
- Format - Format of the created file.
- Database - Defines the commands utilization referring at the database creation.
- Tables - Defines the commands utilization referring at the database creation.
- Data - Defines the commands utilization at the data insertion.

Create Tables

Enables the tables creation through of a simple and intuitive interface, where we can create them quickly.

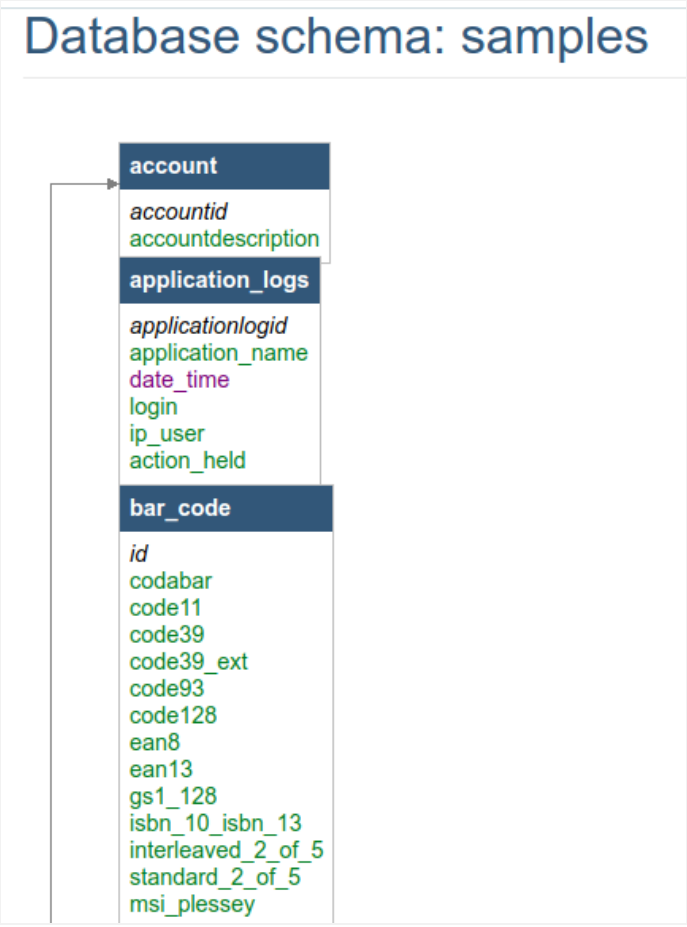


Database Change

Enables change the database name, your default collation or delete the database.

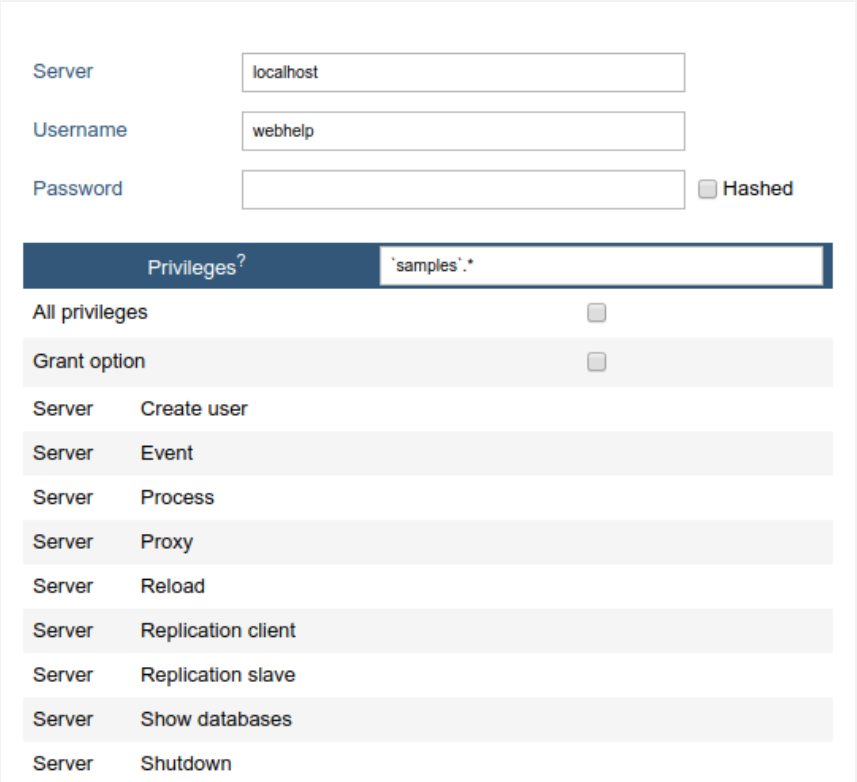
Database Schema

Display a relationship diagram of the database tables.



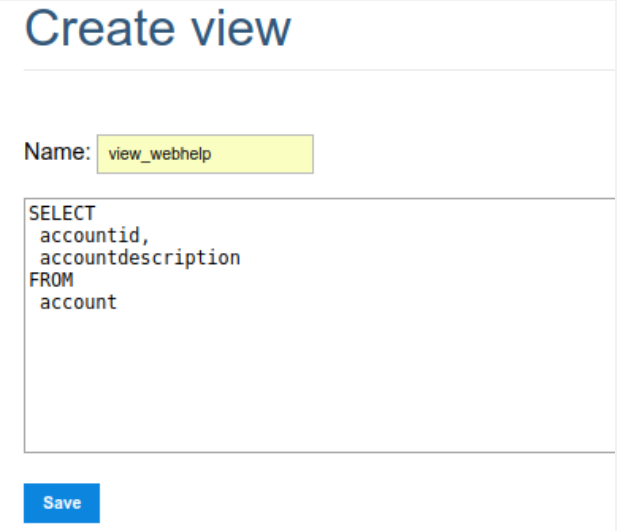
Privileges

Edit the users privileges to the current database.

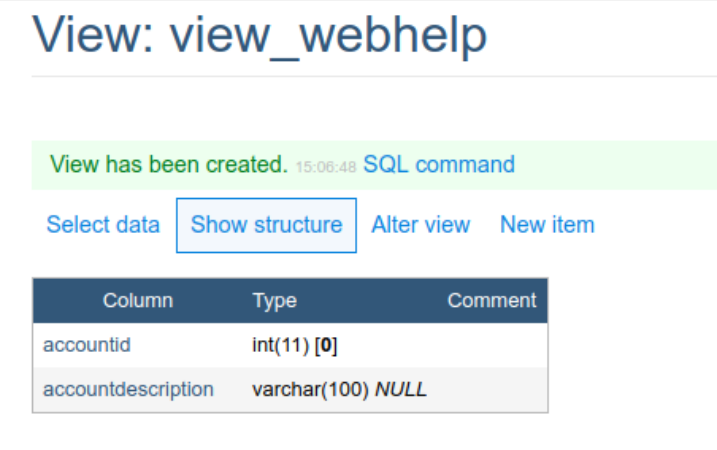


Create View

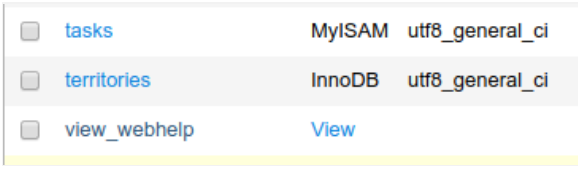
Enables a quickly creation of a view, where we must inform only the name and the select that will be used in the view.



After save, we can see the command executed and the structure of the created view.

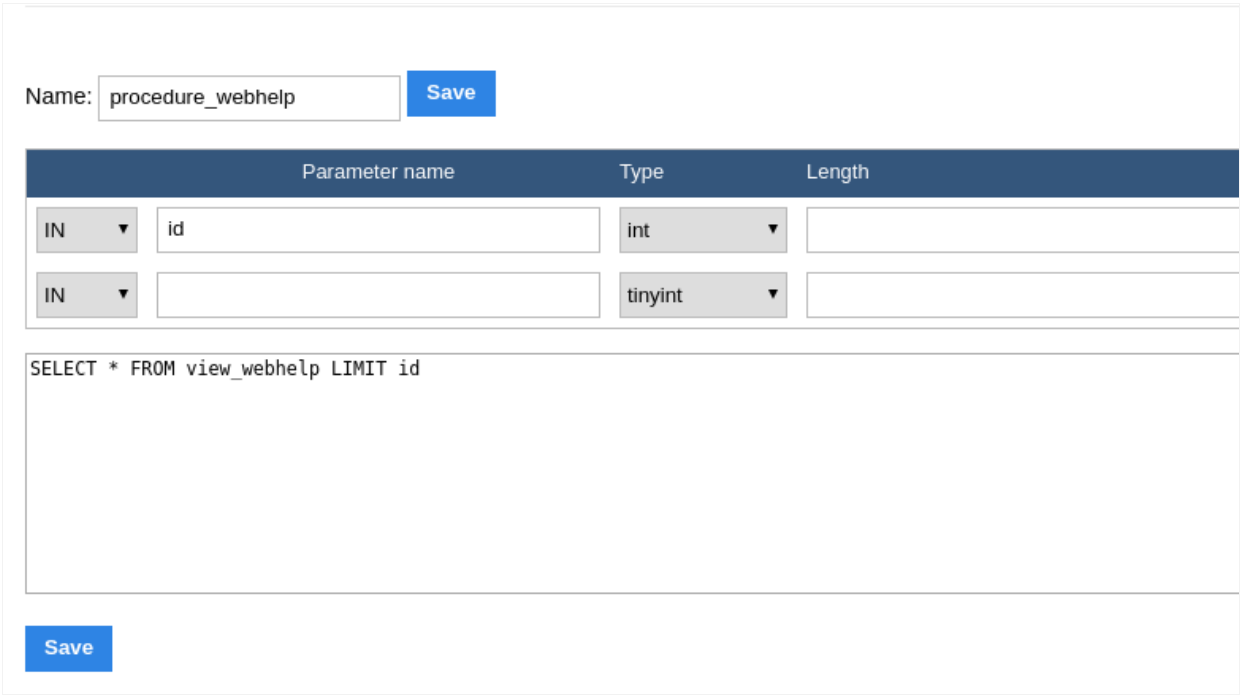


All the created views can be displayed toghether the database tables.



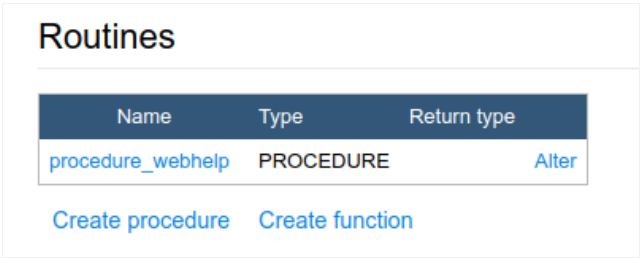
Create Procedure

This functionality makes the procedure creation more easily. The parameters must be informed like in the example below, where we obligatory must inform the parameter type (input and/or output), your name, the data type that will recieve or return and your size.



The executed command will be displayed.

All the created procedures can be visualized below the table and view list.

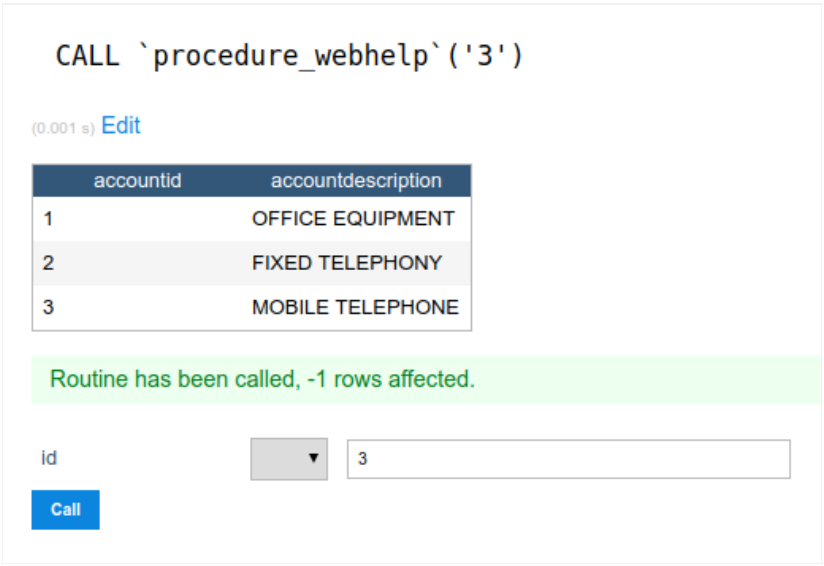


Executing a Procedure

There are two ways to execute a procedure inside the database builder.

At visualize the procedure and functions list, we can click in their name. We will be redirect to the procedure call, where we must inform the input parameter(in case that procedure possess).

After that, will be displayed the result of the procedure execution and the used SQL for that.



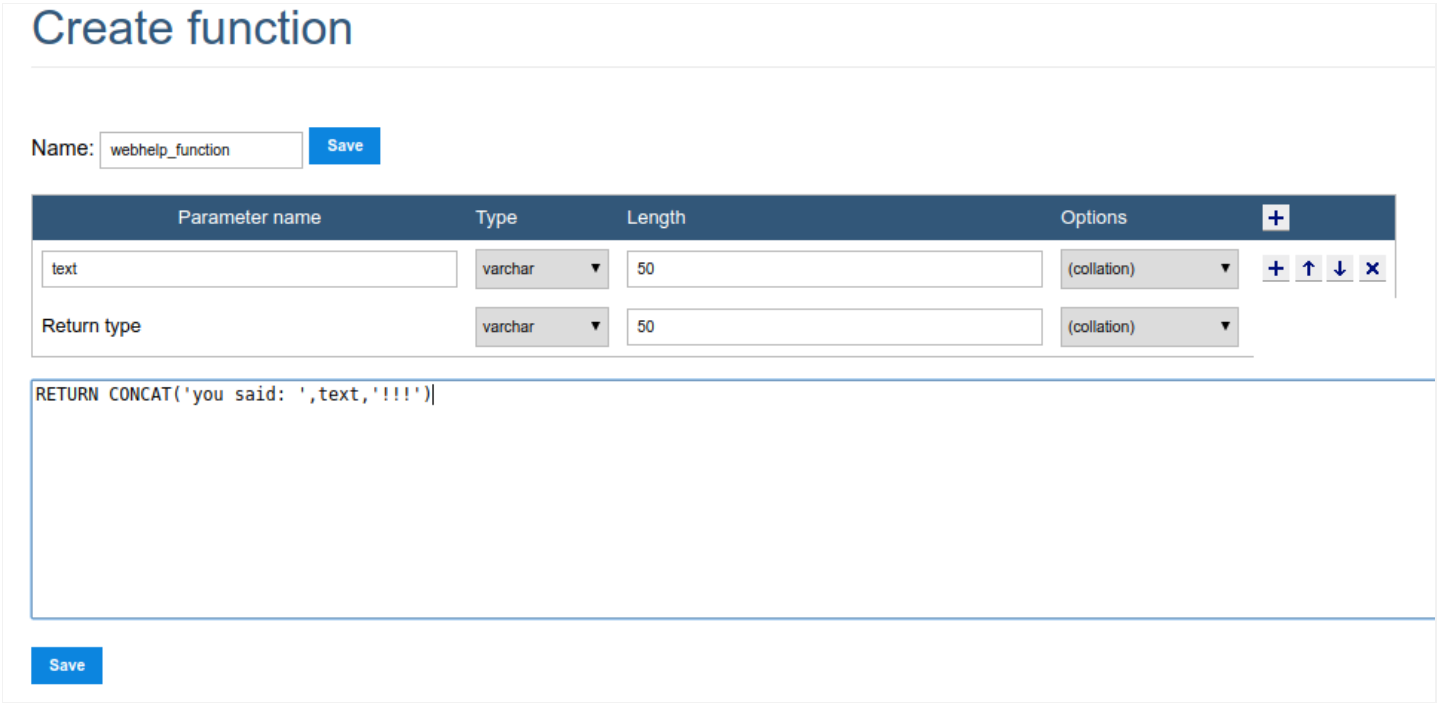
The second way is using the SQL command functionality that allow us to execute any SQL command.

In this case, we must use the command to call a procedure: 'CALL procedure_name(parameter)'.

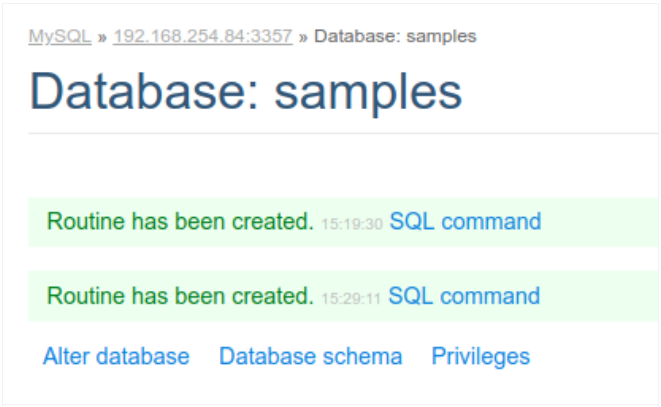
Creating Functions

The parameters must be informed like in the example below, where we obligatory must inform your name, the data type that will receive and the size.

We must define the datatype of the function return too.



The executed command will be displayed.



Executing a Function

There are two ways to execute a function inside the database builder.

At visualize the procedure and functions list, we can click in their name. We will be redirect to the function call, where we must inform the input parameters(in case that possess).

After that, will be displayed the result of the execution and the used SQL for that.

The second way is using the SQL command functionality that allow us to execute any SQL command.

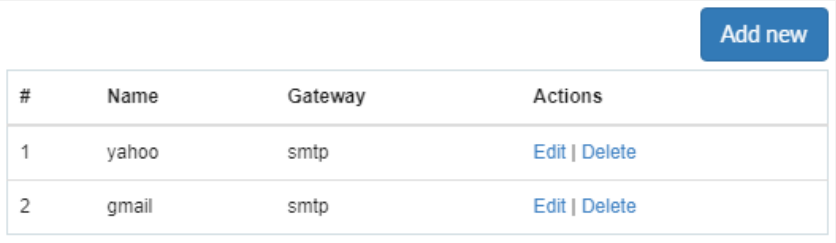
In this case, we must use the command to call a function: 'CALL function_name(parameter)'.

API

API settings let you integrate the various APIs available into the Scriptcase. Through this interface it is possible to configure email sending, SMS and Online Payment APIs.

APIs List

On access you can view a list of all the APIs already configured.



- **Name**: Configured API Name.
- **Getaway**: Getaway used to send the email.

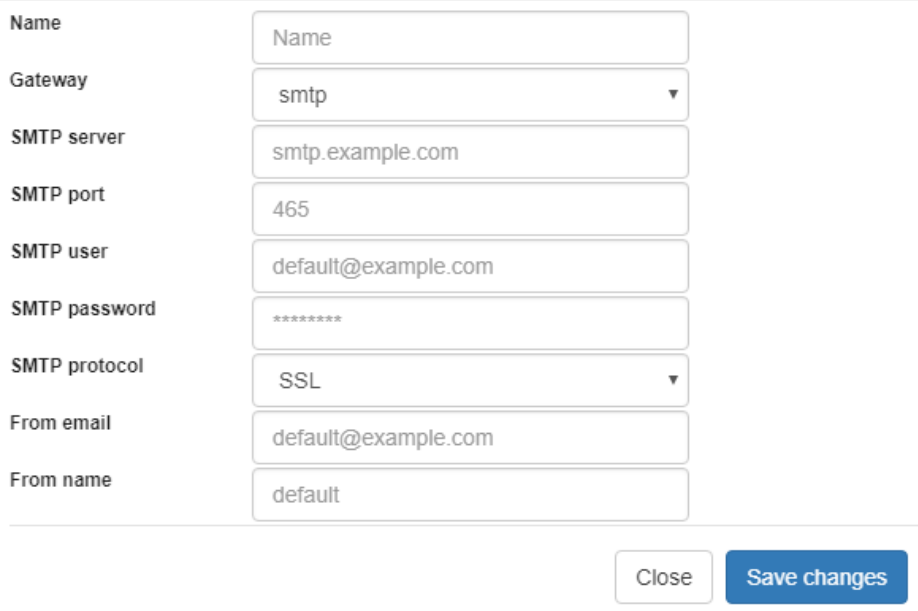
Email APIs

Scriptcase connects to three different APIs for sending emails: SMTP, Mandrill, and Amazon SES.

SMTP Configuration

SMTP is the standard protocol for sending e-mails over the Internet, and each provider has its SMTP.

For more information about SMTP settings click here



API

It allows you to select an API already configured in **Tools > API** or set a new one here by selecting **- custom -**. In this case, you see some options according to the selected Gateway.

To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in

the API menu.

Gateway

Select the way to send the email. The SMTP is the default protocol for sending e-mails over the Internet, and each provider has its SMTP.

SMTP Server

SMTP server address for the provider.

Port SMTP

Port used by the mail server. Use port 465 for security with SSL, port 587 for security with TLS, or port 25 as port without security. By omitting the value, Scriptcase defaults to 25.

User SMTP

SMTP server user.

Enter SMTP

SMTP server user password.

Protocol SMTP

Defines the security protocol. By omitting the value, Scriptcase uses the default value.

E-mail

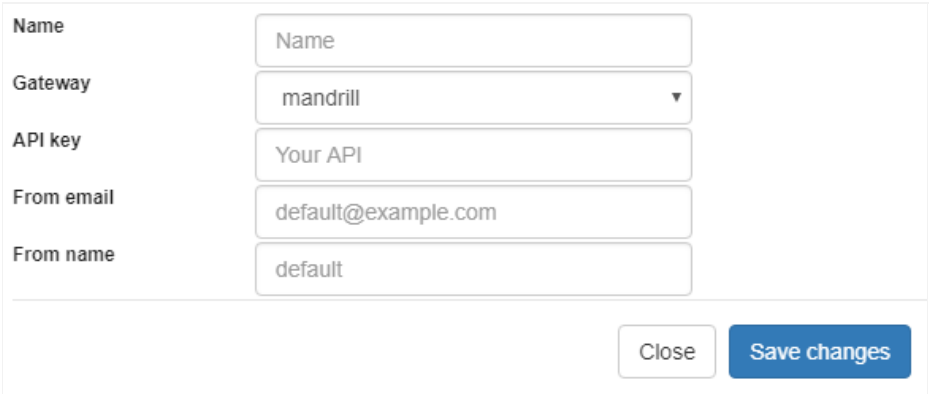
Enter the origin email, that is the email sender.

Name

The sender's name displayed in the email.

Mandrill Configurations

Mandrill is a transactional email API for MailChimp users, ideal for sending data-driven emails.



API

It allows you to select an API already configured in **Tools > API** or set a new one here by selecting **- custom -**. In this case, you see some options according to the selected Gateway.

To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

Gateway

Select the way to send the email. **Mandrill** is a transactional email API for MailChimp users, ideal for sending data-driven emails.

API KEY

Enter the key obtained from the origin site of your API.

E-mail

Enter the origin email, that is the email sender.

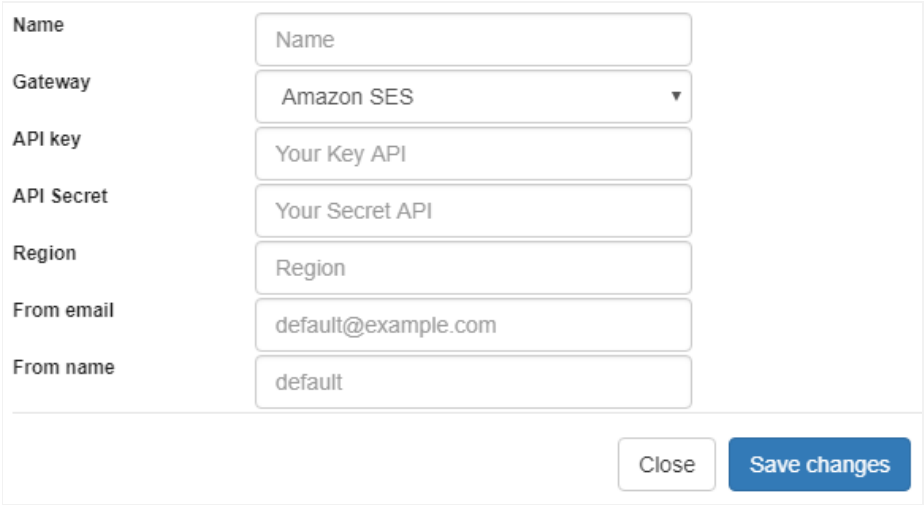
Name

The sender's name displayed in the email.

For more information about mandrill

Amazon SES Configurations

Amazon Simple Email Service Amazon SES is an email sending service designed to assist in sending marketing emails, notifications, and transactional messages.



To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

API KEY

Enter the key obtained from the origin site of your API.

API Secret

Enter the secret access key of your account.

Region

Amazon SES has endpoints in many countries, to reduce network latency, inform the region of the endpoint closest to your application. See the regions.

E-mail

Enter the origin email, that is the email sender.

Name

The sender's name displayed in the email.

For more information about Amazon SES

SMS APIs

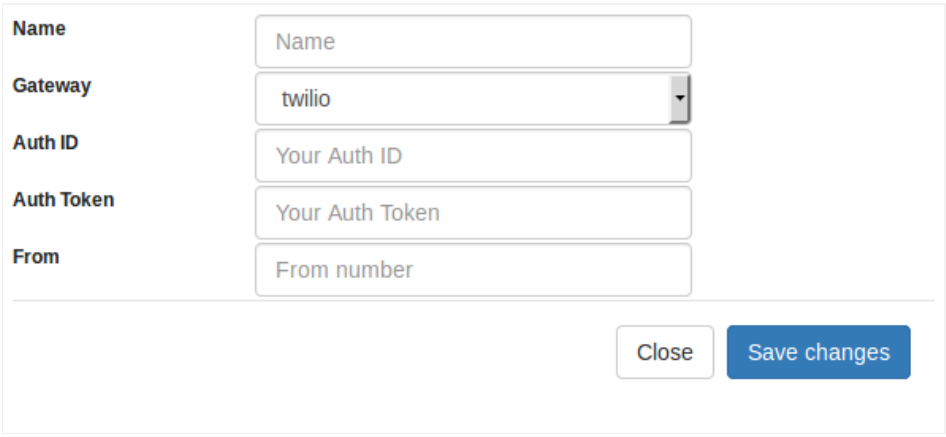
Scriptcase connects to different APIs for sending SMS text messages:

- Twilio
- Plivo
- Clickatell

Watch a video of how Sending SMS notifications

Setting up Twilio API

Twilio is a set of libraries that enables SMS communication on a global scale. For correct configuration, we need to enter the data of **Authentication ID**, **Authentication Token**, and **Sender**, as shown below:



To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

Name

Enter a name for your API;

Gateway

The Gateway we want to use. (in this case, Twilio);

Auth ID

Enter the authentication ID for your Twilio account;

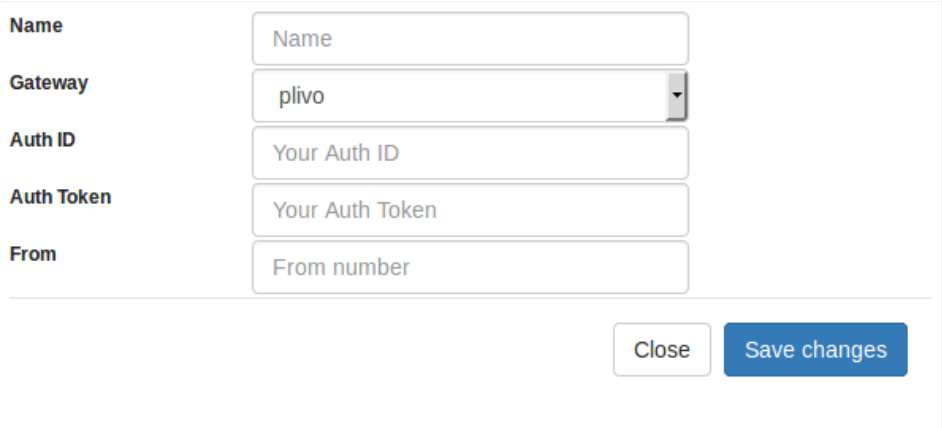
Auth Token

Enter the authentication Token for your Twilio account;

From

Enter the sender's number to send the SMS;

Setting up Plivo API



To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

Name

Enter a name for your API;

Gateway

The Gateway we want to use. (in this case,Plivo);

Auth ID

Enter the authentication ID of your Plivo account;

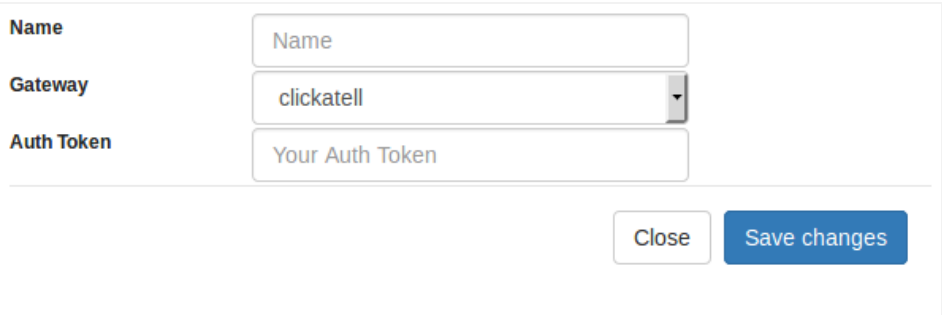
Token ID

Enter the authentication Token of your Plivo account;

From

Enter the sender number to send the SMS;

Setting up Clickatell API



To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

Name

Enter a name for your API;

Gateway

The Gateway we want to use. (in this case, Clickatell);

Auth Token

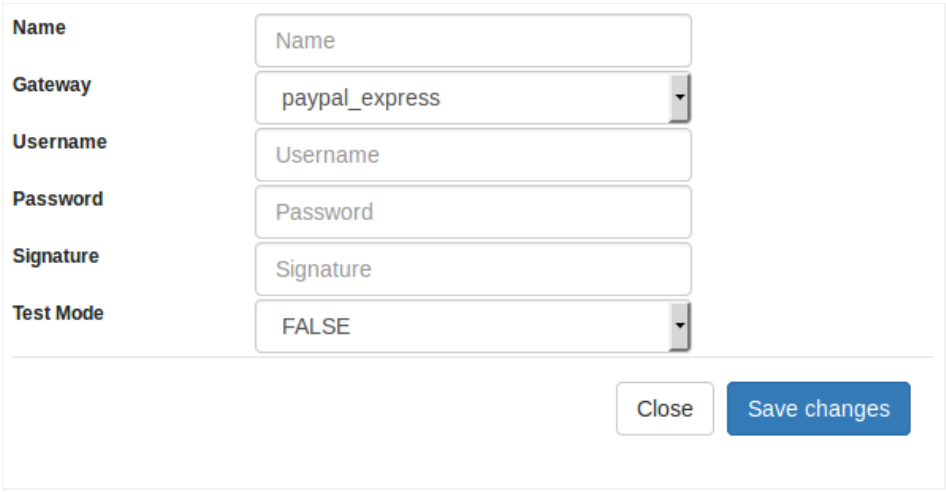
Enter the authentication Token of your Clickatell account;

Payment APIs

Scriptcase allows using APIs for online payment processing. The following API integrations are available:

- paypal
- Mercado Pago
- Braintree
- Stripe

Setting up PayPal



To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

Name

Enter a name for your API;

Gateway

Enter the Gateway to use (For example, paypal_express);

Username

Enter the user of PayPal account for authentication;

Password

Enter the password of PayPal account for authentication;

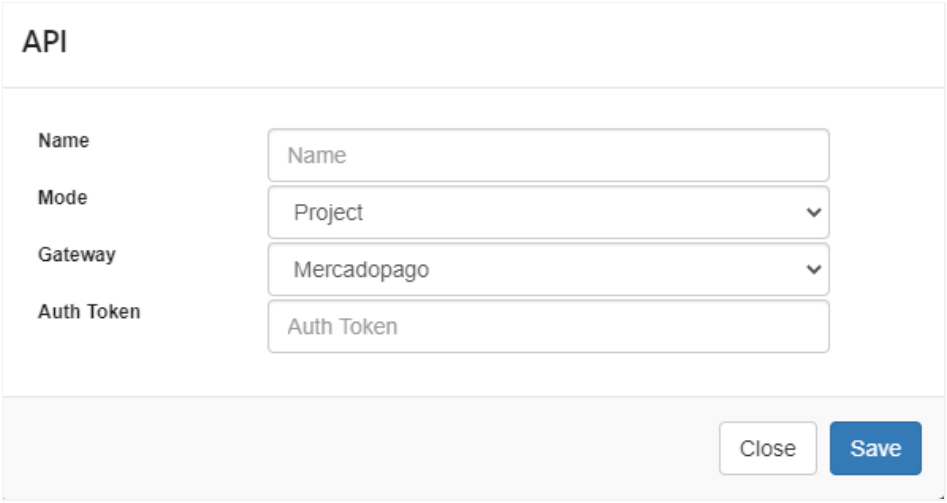
Signature

Enter the signature of PayPal account for authentication;

Test Mode

It allows you to use the API as a test mode.

Configuring MercadoPago payment



Available in the following countries:

- Argentina
- Brazil
- Chile
- Colombia
- Mexico
- Peru
- Uruguay

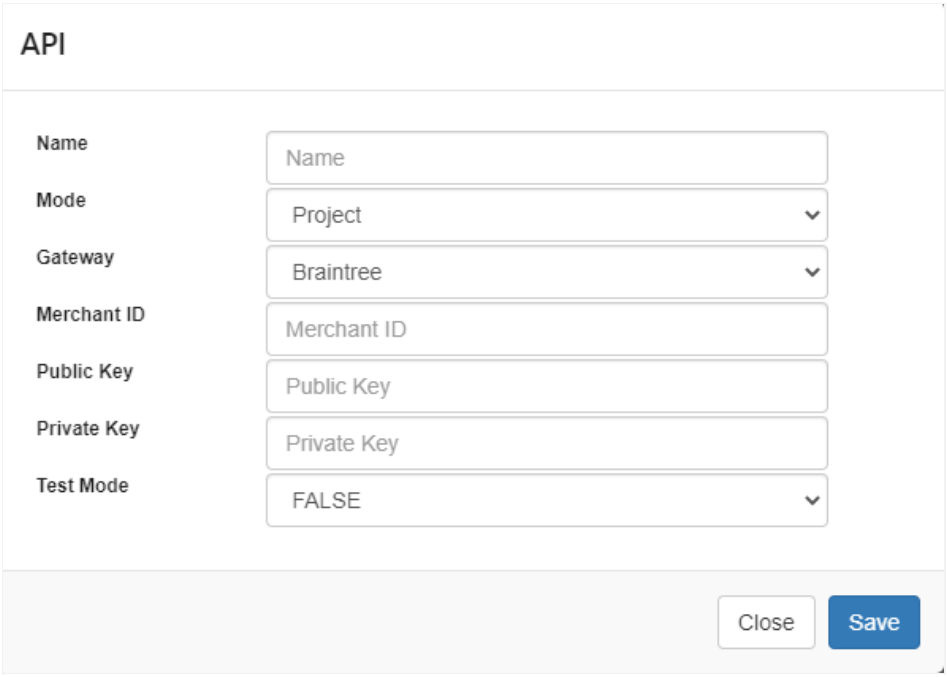
To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

Mercadopago;

- **Name**: Name given to the API;
- **Modo**: Defines whether the API created will be at the Project, User or Public level;
- **Gateway**: The **Mercadopago**;
- **Auth Token**: Application authentication token.

To learn how to create Mercado Pago credentials, follow this tutorial: [Creating your Mercado Pago credentials] [Mercadopagotutorial]

Configuring payment by Braintree



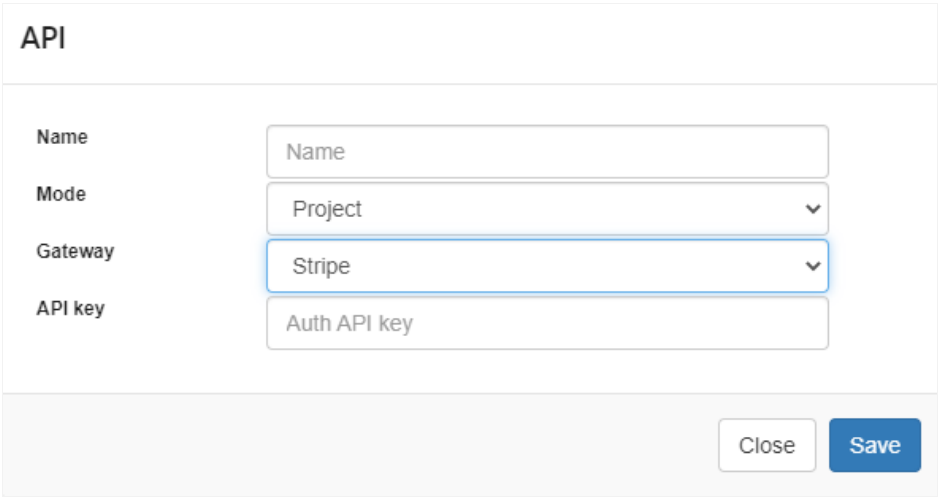
To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

- **Name**: Defines the name that will identify the configured API.
- **Modo**: Defines the access level of this API, which are: Public, where all users of all projects have access to this API; Project, where all users of this project will have access to this API; User, where only the API creator user will have access in any project. (Advisable for this API at the project level).
- **Gateway**: Defines the API that will be used. We must select Braintree.
- **Marchant ID** - Your Merchant ID is the unique identifier for your entire gateway account, including the various merchant accounts that may be on your gateway. Often called a public ID or production ID, your Merchant ID will be different for your production and sandbox gateways.
- **Publick Key** - This is your user-specific Public Id. Each user associated with your Braintree gateway will have its own public key.
- **Private Key** - This is your user-specific Private Key. Each user associated with your Braintree gateway will have its own private key. Your private key must not be shared outside of using an API call – even with us.
- **Test Mode** - Enables or disables the API test mode.

Example informing the profile of the API created in the interface as a parameter in the macro

```
$gateway = sc_call_api('braintree');

$response = $gateway->transaction()->sale([
  'amount' => '10.00',
  'paymentMethodNonce' => $nonceFromTheClient,
  'deviceData' => $deviceDataFromTheClient,
  'options' => [
    'submitForSettlement' = > True
  ]
]);
echo '<pre>';
print_r($response);
```

Configuring payment by Stripe



To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

- **Name**: Defines the name that will identify the configured API.
- **Modo**: Defines the access level of this API, which are: Public, where all users of all projects have access to this API; Project, where all users of this project will have access to this API; User, where only the API creator user will have access in any project. (Advisable for this API at the project level).
- **Gateway**: Defines the API that will be used. We must select Braintree.
- **API key** - Enter the API key.

Example informing the profile of the API created in the interface as a parameter in the macro

```
$gateway = sc_call_api('stripe');

$gateway->charges->create([
  'amount' => 2000,
  'currency' => 'brl',
  'source' => 'tok_mastercard',
  'description' => 'My First Test Charge with Scriptcase',
] );
```

Storage APIs

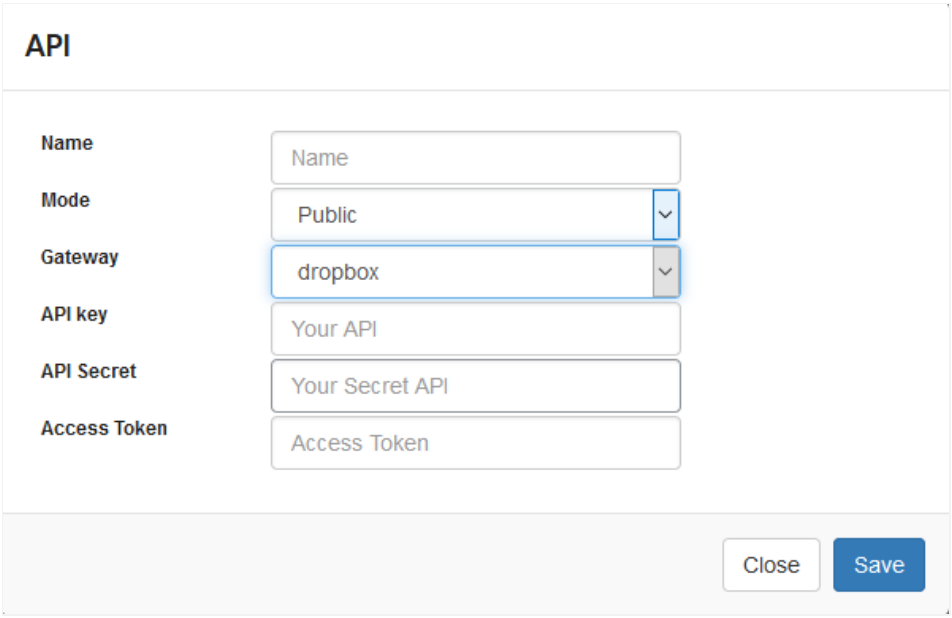
The scriptcase provides 3 APIs for this type of service. It only works in fields of type: image (image file name) and document (document file name).

- **Dropbox**
- **Google Drive**
- **S3**

Configuring the dropbox API

Dropbox is a service for storing and sharing files. It is based on the concept of “cloud computing”. It belongs to Dropbox Inc., based in San Francisco, California, USA. The company that developed the program provides computer centers that store its customers’ files.

To learn how to configure your API key click here



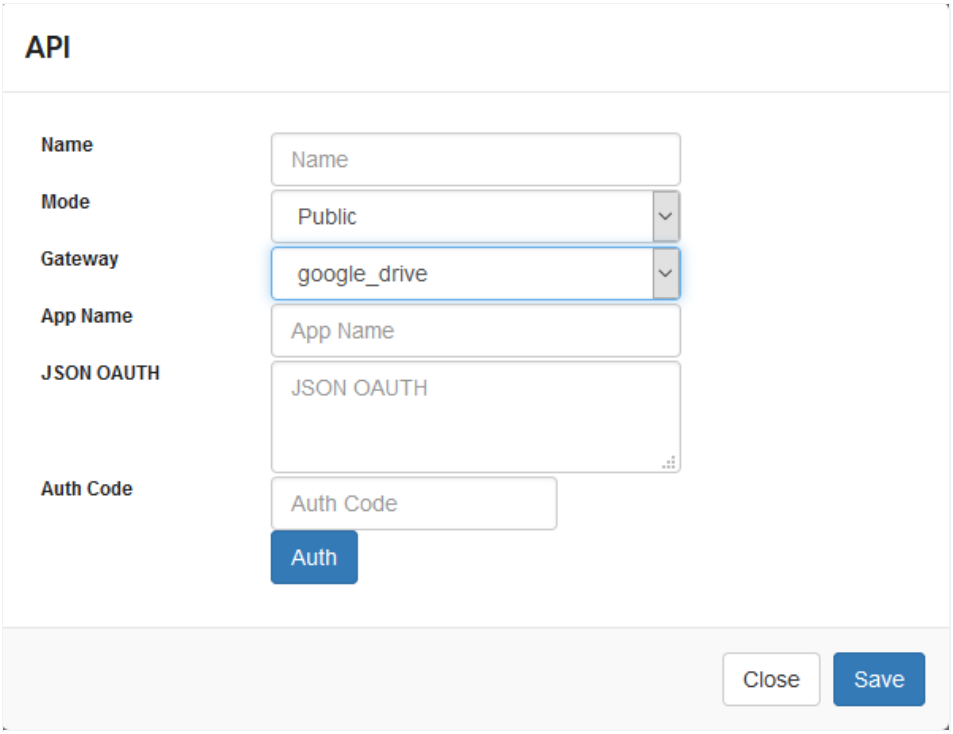
To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

- **Name**: Defines the name that will identify the configured API. We recommend that the name be defined in a way that allows identifying the data used in the configuration, since it is possible to create more than one API with this Gateway, this way it will be simpler to identify the API at the time of the call.
- **Mode**: Defines the access level of this API, which are: Public, where all users of all projects have access to this API; Project, where all users of this project will have access to this API; User, where only the user who created the API will have access to any project. (Advisable for this API the project level).
- **Gateway**: Defines the API to be used. We must select Dropbox.
- **API Key**: Enter the key obtained when configuring your API.
- **API Secret**: Enter the secret access key for your account.
- **Access Token**: Enter the access Token generated in your Dropbox account.

Google Drive

Google Drive is a virtual disk service launched by Google. The service allows the storage of files in the Google cloud.

To learn how to configure your API key click here



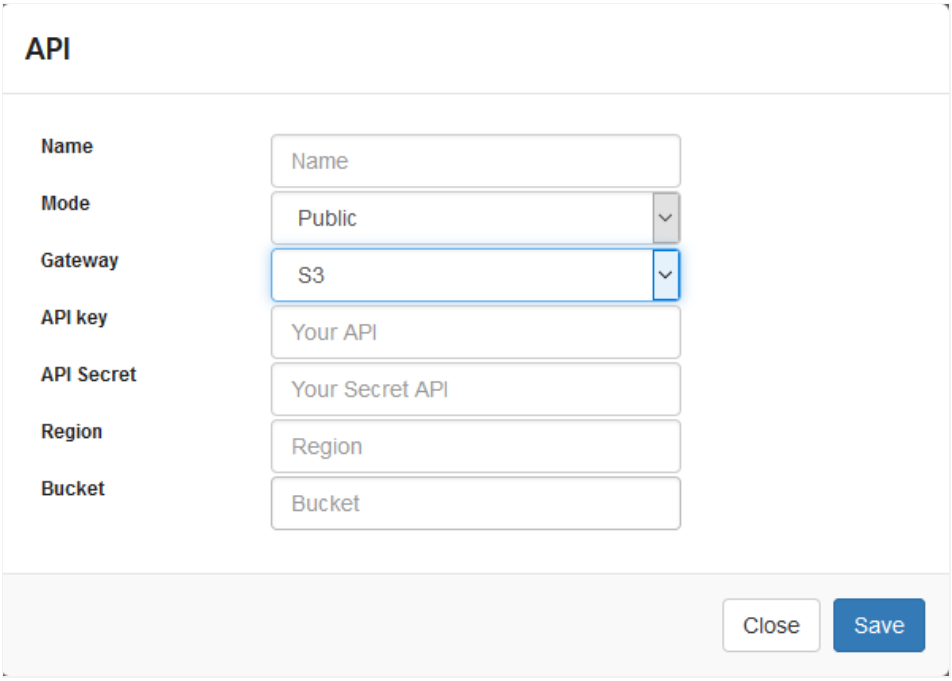
To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

- **Name**: Defines the name that will identify the configured API. We recommend that the name be defined in a way that allows identifying the data used in the configuration, since it is possible to create more than one API with this Gateway, this way it will be simpler to identify the API at the time of the call.
- **Mode**: Defines the access level of this API, which are: Public, where all users of all projects have access to this API; Project, where all users of this project will have access to this API; User, where only the user who created the API will have access to any project. (Advisable for this API the project level).
- **Gateway**: Defines the API to be used. We must select Google drive.
- **Application Name**: Name of the application to be used.
- **Json OAuth**: Enter the contents of the generated user authentication file.
- **authentication code**: Enter the identification code obtained when configuring your API.

Amazon S3

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security and performance. This means that customers of all sizes and sectors can use it to store any amount of data in a wide variety of use cases, such as websites, mobile applications, backup and restore, archiving, business applications, IoT devices and analytics of big data.

To learn how to configure your API key click here



To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

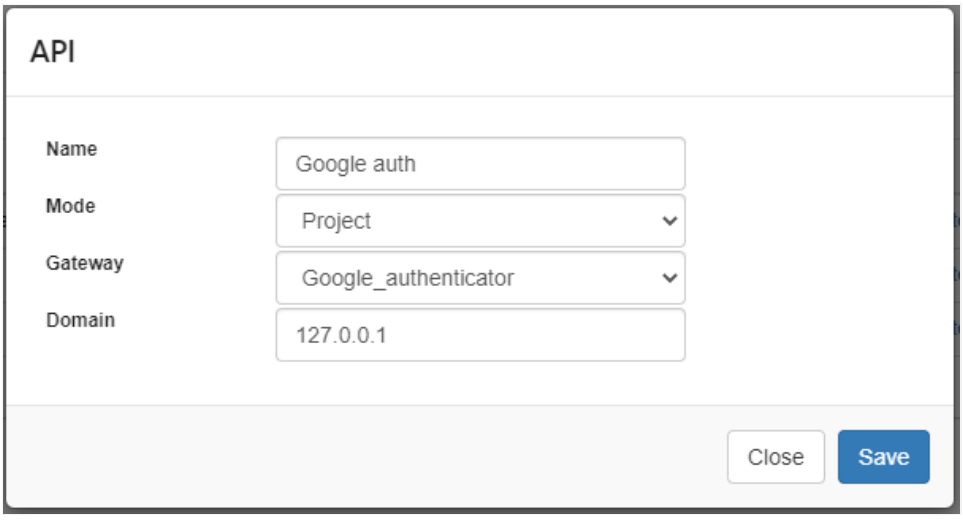
- **API Key**: Enter the key obtained when configuring your API.
- **API Secret**: Enter the secret access key for your account.
- **Region**: Amazon SES has endpoints in several regions, in order to reduce network latency, inform the endpoint region closest to your application.
- **Bucket**: Inform the bucket to store files on Amazon S3.

Google Authenticator API

This google authentication API allows you to create more secure systems using two-factor authentication.

To use this API it is necessary to install the **Google Authenticator** application on your mobile device.

- Google Store
- Apple Store



To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in

the API menu.

For API configuration We must inform:

Name

Used only for API identification within the tool.

Mode

This option defines the availability of the API within Scriptcase. The configuration options are:

- **Public** - With this option all projects and users of the tool will have access to this API for use.
- **Project** - When saving the API at the project level, all users of the project will have access to this API for use.
- **User** - Only the user who performed the API configuration will have access to its use in any project.

The resource can only be adopted in the project where it was created, regardless of the chosen save mode.

Gateway

Defines the authentication API that will be used.

Domain

Defines the domain enabled to use two-factor authentication.

Examples:

- myscriptcase.com.br
- 127.0.0.1

We must inform only one domain to use the API.

Google Sheets API

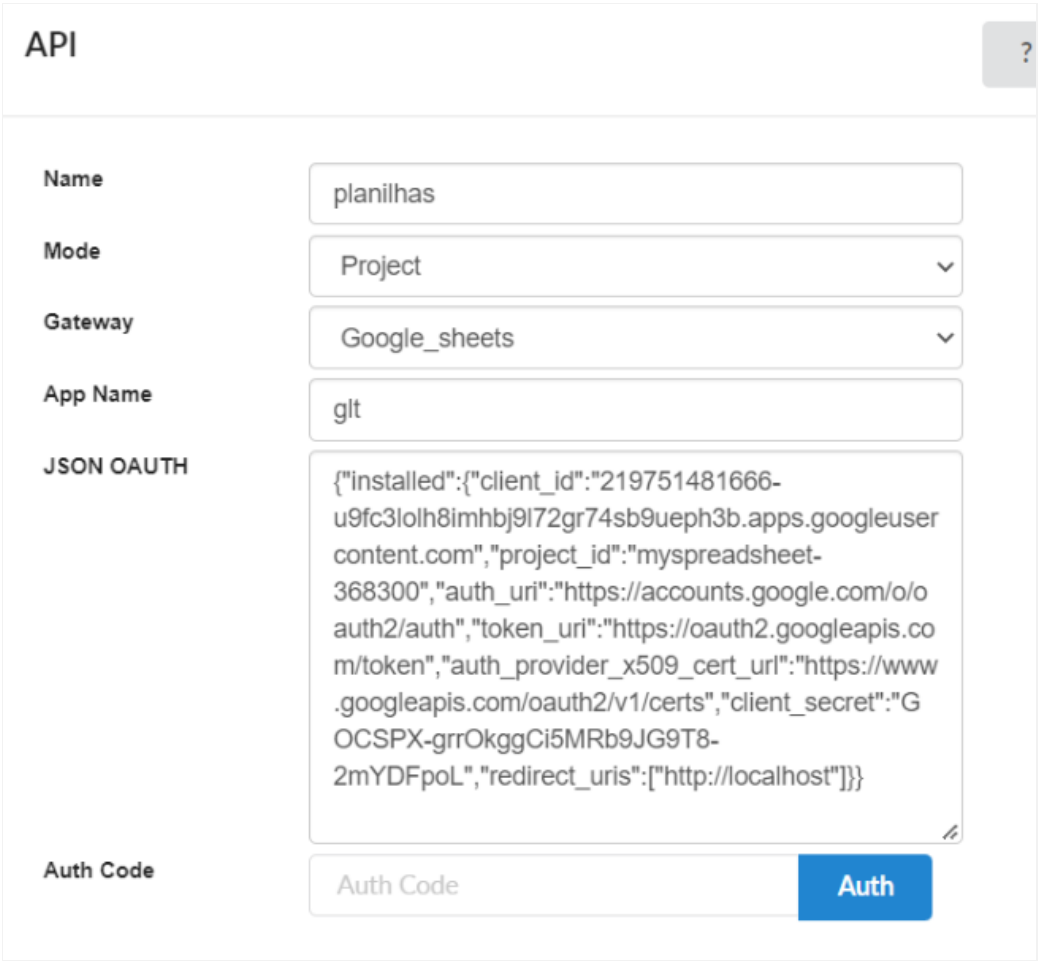
Scriptcase allows using google sheets API for integration with your projects.

Google Sheets

Google Sheets is a spreadsheet program included as part of the free web-based Google Doc Editor suite offered by Google.

To learn how to create and configure your credentials click here

This API has the functionality of just inserting data in the spreadsheet. There is no possibility of deleting or editing spreadsheet records.

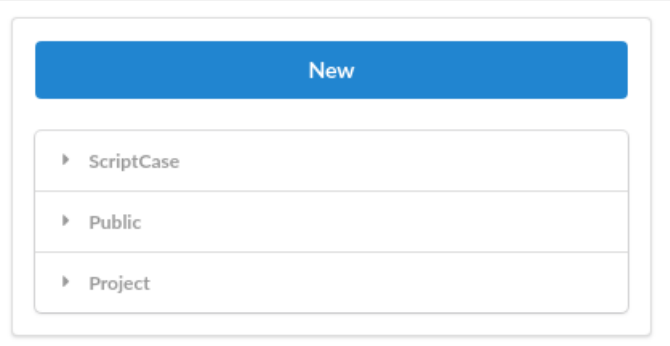


To use an API in a published project, it is necessary to perform the Configuration of this API in the production environment, in the API menu.

- **Name**: Defines the name that will identify the configured API. We recommend that the name be defined in a way that allows identifying the data used in the configuration, since it is possible to create more than one API with this Gateway, this way it will be simpler to identify the API at the time of the call.
- **Mode**: Defines the access level of this API, which are: Public, where all users of all projects have access to this API; Project, where all users of this project will have access to this API; User, where only the user who created the API will have access to any project. (Advisable for this API the project level).
- **Gateway**: Defines the API to be used. We must select Google drive.
- **Application Name**: Name of the application to be used.
- **Json OAuth**: Enter the contents of the generated user authentication file.
- **authentication code**: Enter the identification code obtained when configuring your API.

Hotkey Templates

This tool allows that the user manage templates to the application hotkeys. Can be possible create a new hotkeys set or use an existing setting and change then to improve your development environment.



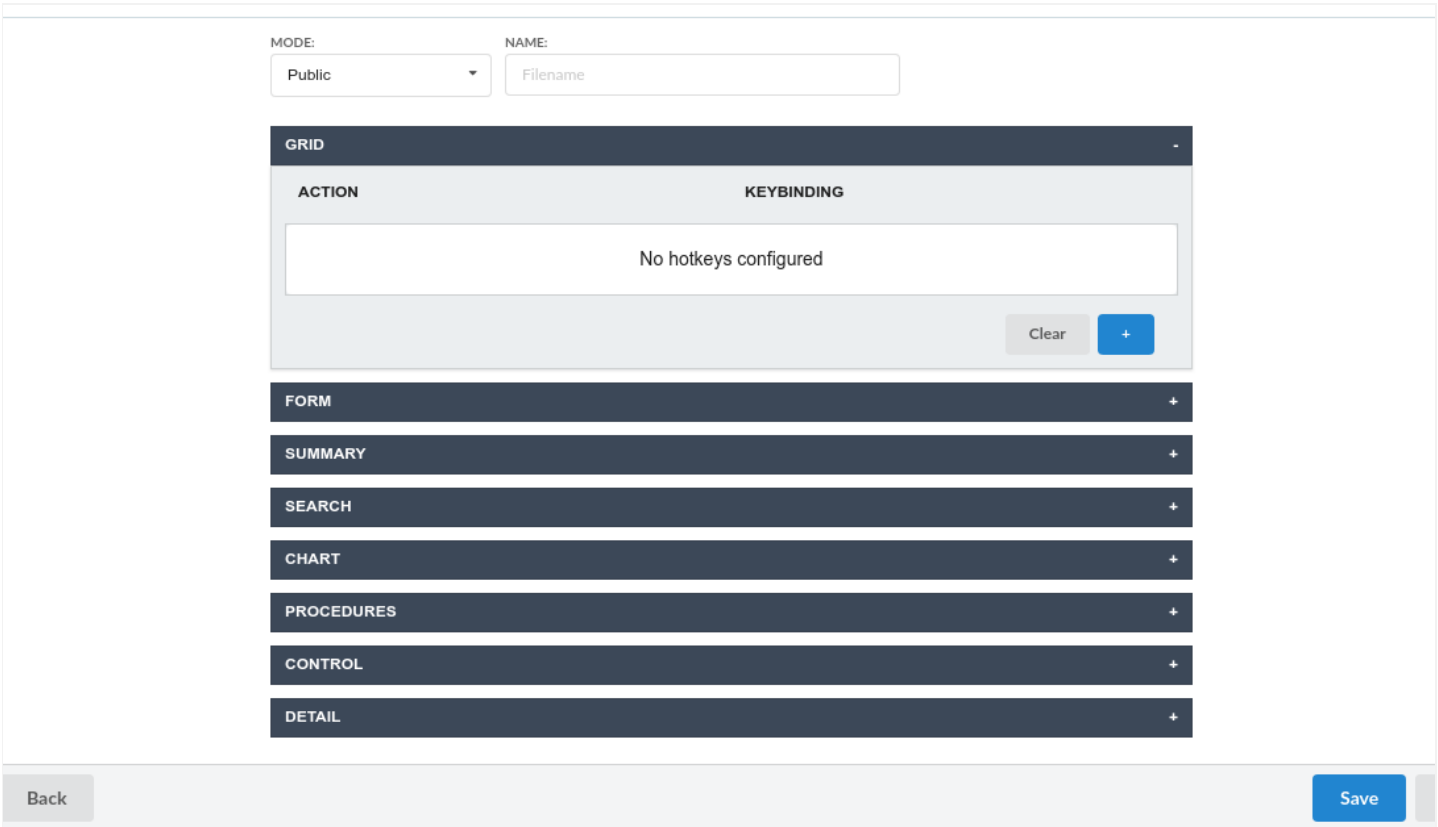
Home screen to the template and scopes managements available.

Available Scopes:

- **Scriptcase**: Here is the default Scriptcase template. This template cannot be edited or erased, only can visualized or a copy can be made to utilize the already defined hotkeys.
- **Public**: Here is the created and edited templates at public level by user.
- **Project**: Here is the created and edited templates at project level by user.

In “New” the user will can create a new template with all the configurable hotkeys.

Then, will come the creation screen, when the user will can select the name and the scope of the template that will be created.

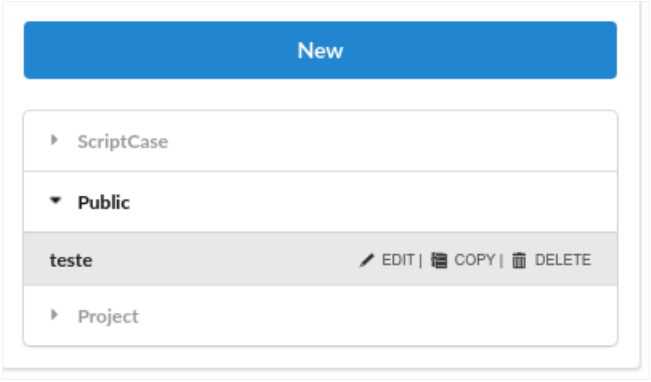


Creation screen of Hotkey Templates

- **Mode**: Allows the user to select the scope that the template will be created.

- **Name**: Allows the user to select the name to the template that will be created.
- **Action**: Selects the action that will be made when press the selected keybind.
- **Keybinding**: Selects the keybinds that will be responsible for execute the chosen action.
- **Add "+"**: Adds a new action in the keybind list.
- **Clear**: Clears all predefined settings by Scriptcase or by user.

When selecting one of the categories that possess created templates, the user will have some setting options. All the applications will come with the Scriptcase default template configured.



Options to Template configuration.

- **Edit**: Allows the user to edit the name, mode and the defined hotkeys in the template.
- **Copy**: Creates a copy of the selected template allowing to use the same configurations, change the name and hotkeys settings that the user desires.
- **Delete**: Realizes the exclusion of the selected template.

When editing the name or the scope of a template, automatically all the related applications to them will lose your own configurations. Will be necessary configure the edited template again as in use in the application.

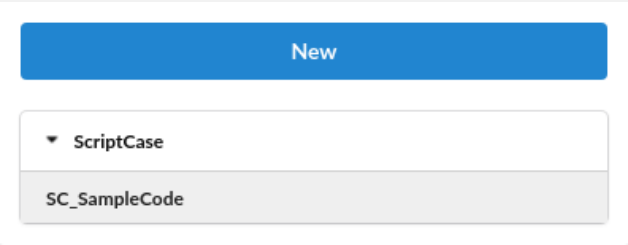
Code Snippets

This tool allows the developer to perform the creation of blocks of code and also to use these codes easily in Scriptcase applications. In this way, streamlining and facilitating the use of programming within the tool.

The created codes can be used in all events present in Scriptcase applications. Through the **Snippets** tab.

Creating a Code Snippet

To create a Code Snippet, simply click the **New** button available in the interface.



View:

It allows the user to view all the code, but without the possibility to make edits.

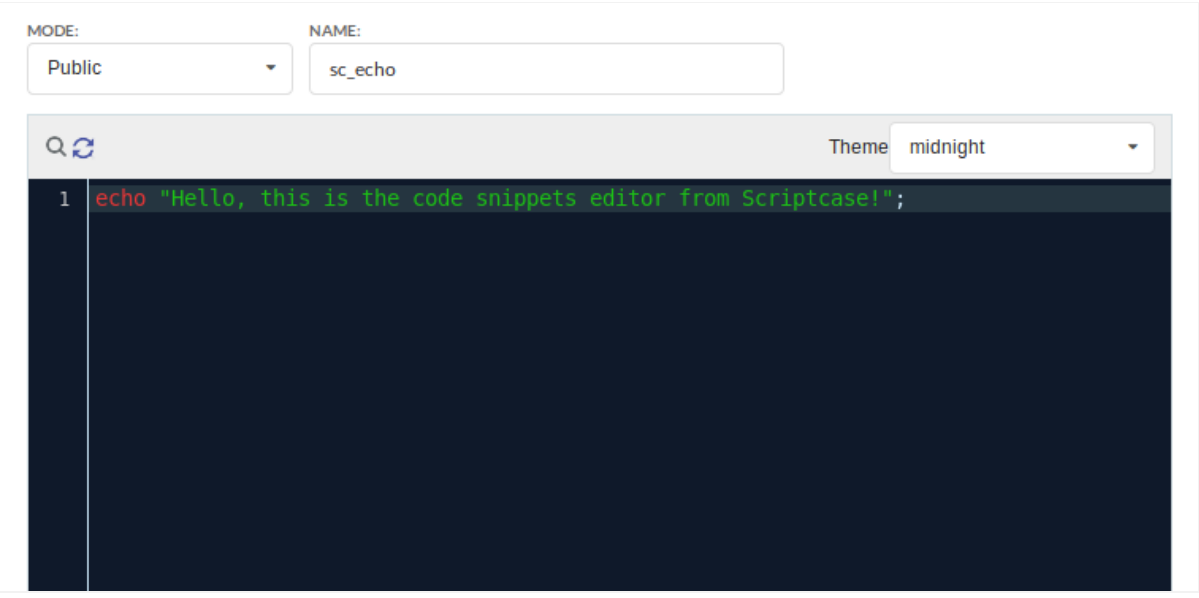
Copy:

It makes a copy of the code snippet for a new one, allowing the user to change where it will be located and the usage name.

Edit:

It allows the complete editing of the selected code, as well as where it is located and its name of use. This option is only available for the code snippets located in Public, Project, and User.

After that, the code edit screen will open, where you can create the code desired by the developer. See the image below:



Mode:

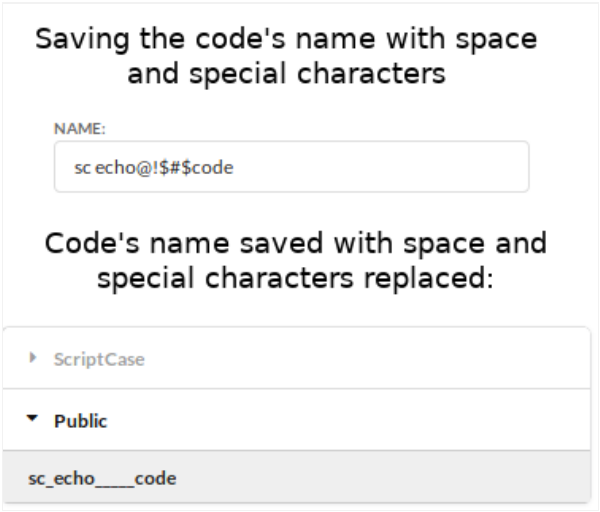
Lets you select where the code will be created. As available options there are: Public, Project and User.

Name:

Lets you define the name that will be used to call the code in the event.

Any special character and/or space that is entered in the name is automatically replaced by the symbol (_).

Example:



Themes:

Allows you to select the theme used in the code editor. If you already have a previously selected theme for editing an event, the same theme is applied to the code snippet editor.

Search:

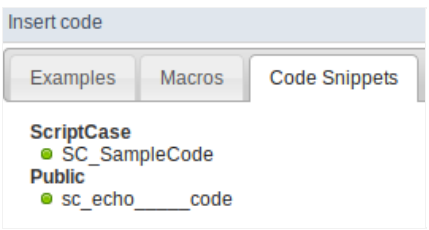
The search icon allows the user to search for one or more terms within the code.

Replace:

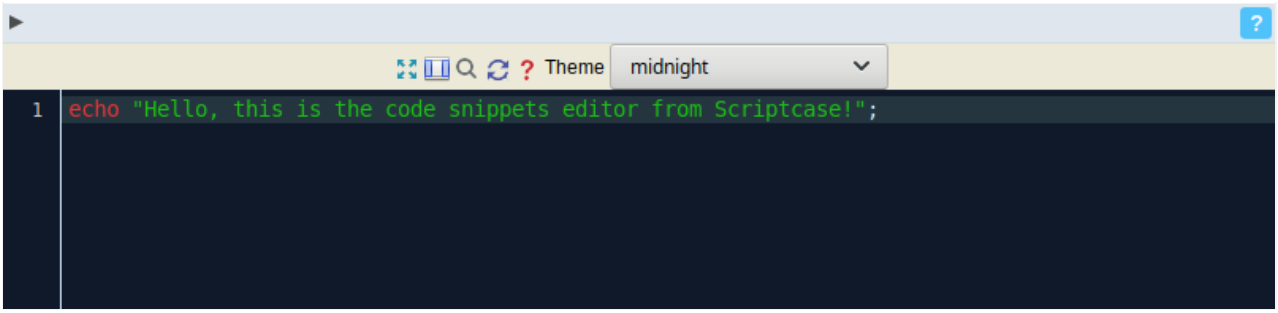
The replace icon allows you to replace snippets of code with another. Thus, facilitating the editing of several terms within the code.

Using a Code Snippet

To use a code snippet you must have access to any event present in the Scriptcase. By accessing the **Snippets** tab, you can view the list of code snippets created by Scriptcase and the user.



When you click on the usage name, the code is automatically added to the event.


Theme
midnight
1
echo "Hello, this is the code snippets editor from Scriptcase!";

SMTP Settings

To proceed with the password recovery, it is necessary that has the SMTP server configured for a successful process. Look here how to realize this procedure.

Yahoo

For more information about these settings Click Here.

Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link. See also:

Ways to securely access Yahoo Mail

SMTP Server Settings Yahoo	
SMTP Server	smtp.mail.yahoo.com
Port	465 or 587
E-mail address	Your full email address (name@domain.com.)
Password	The password for your account.
Requires SSL	Yes
Requires TSL	Yes (if available)
Requires authentication	Yes

Gmail

For more information about these settings Click Here.

Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.

SMTP Server Settings Gmail	
SMTP Server	smtp.gmail.com
E-mail address	Your full email address (Ex. example@gmail.com)
Password	The password for your account.
Port (TLS)	587
Port (SSL)	465
TLS / SSL Required	Yes

Outlook / Hotmail

For more information about these settings Click Here

SMTP Server Settings Outlook.com	
SMTP Server	smtp-mail.outlook.com
E-mail address	Your full email address (Ex. myname@outlook.com, not an alias)
Password	The password for your account.
Port	587 (you can use port 25 as an alternative)
TLS / SSL Required	Yes

Layout Overview

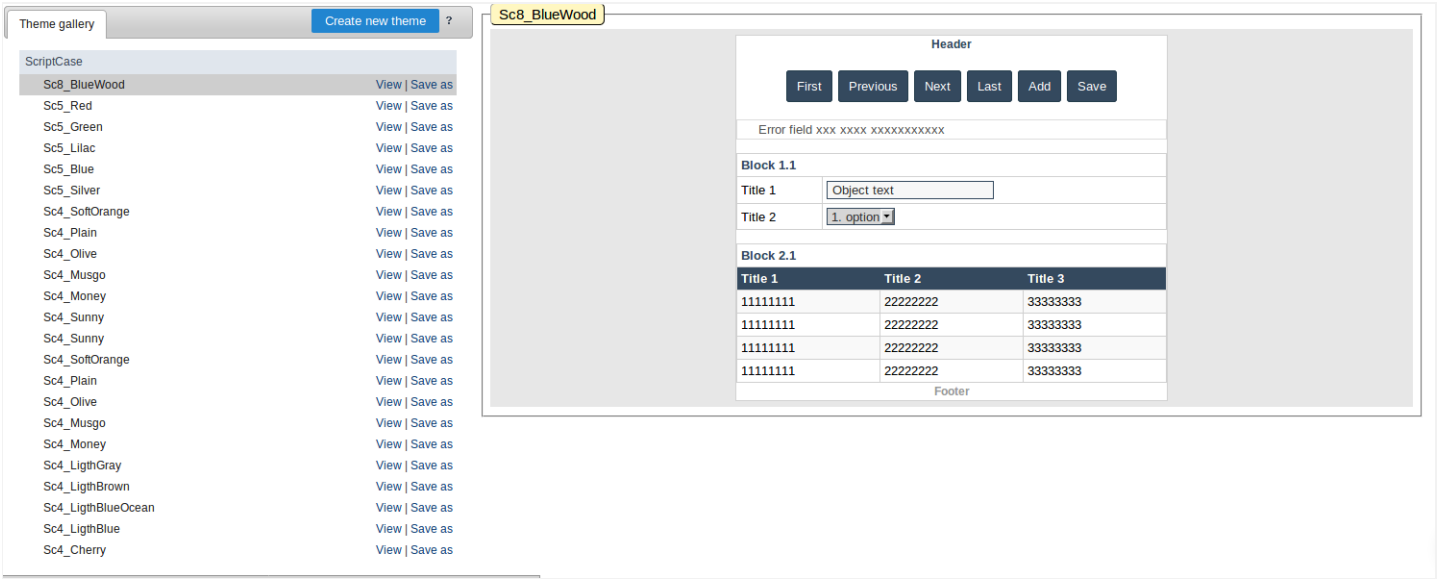
In the **Layout** menu of Scriptcase, you have various options to customize the appearance and behavior of your applications. Below is an overview of each available item:

- **Application Themes**: Allows you to create or edit themes that define the color schemes, fonts, and other visual styles of the applications.
- **Button Themes**: Enables the customization of the styles of buttons used in the applications, including colors, sizes, and icons.
- **HTML Templates**: Offers the option to create or modify HTML templates to be used in the applications, allowing for more advanced layout customization.
- **CSS Menus (For older menus)**: Intended for editing the CSS style of traditional menus, allowing adjustments to the appearance of classic menus.
- **Responsive Menu Themes**: Facilitates the creation and editing of specific themes for responsive menus, adjusting the color palette and styles for better adaptation on mobile devices.
- **Tooltip Themes**: Allows customization of the appearance of tooltips displayed in the applications, adjusting colors and styles as needed.
- **Image Manager**: A tool to manage the images used in the applications, including uploading, organizing, and deleting image files.
- **Chart Themes**: Enables the creation and editing of specific themes for charts, allowing customization of colors, fonts, and other visual elements of the charts generated in the applications.
- **HTML Editor Templates**: Offers the option to create or edit templates for the integrated HTML editor, allowing customization of the editable content in the applications.

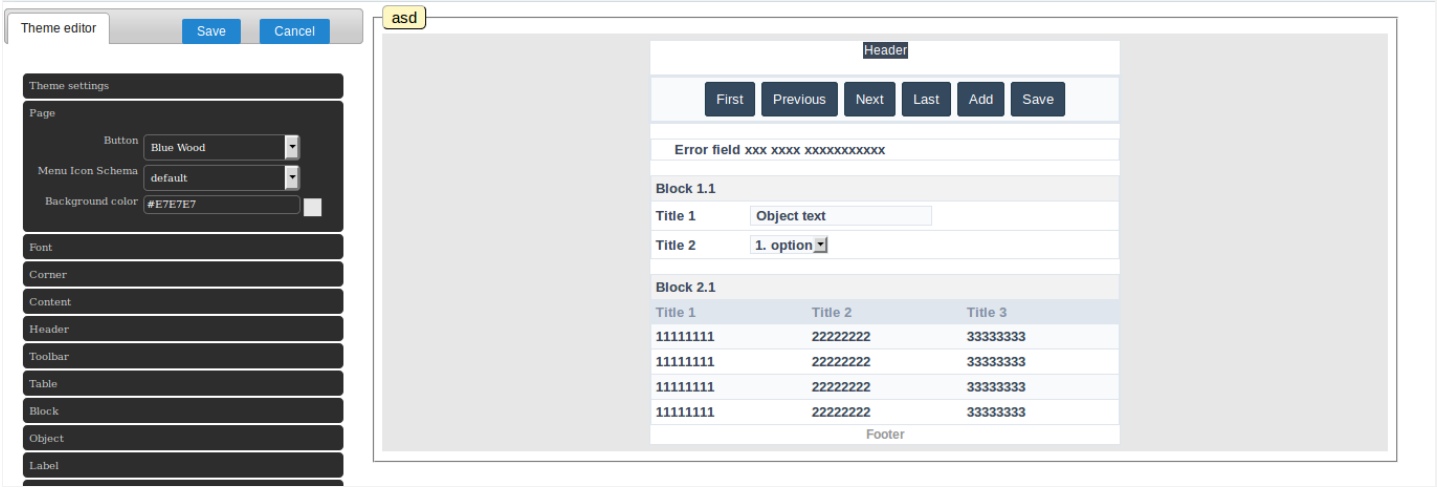
These tools provide flexibility in customizing the layout and style of your applications in Scriptcase, allowing you to adapt the interface to the specific needs of your project.

Scriptcase Applications themes

In the Applications themes section, it's possible to create layouts for the applications using the editor. It's possible to create new themes or edit existing ones.



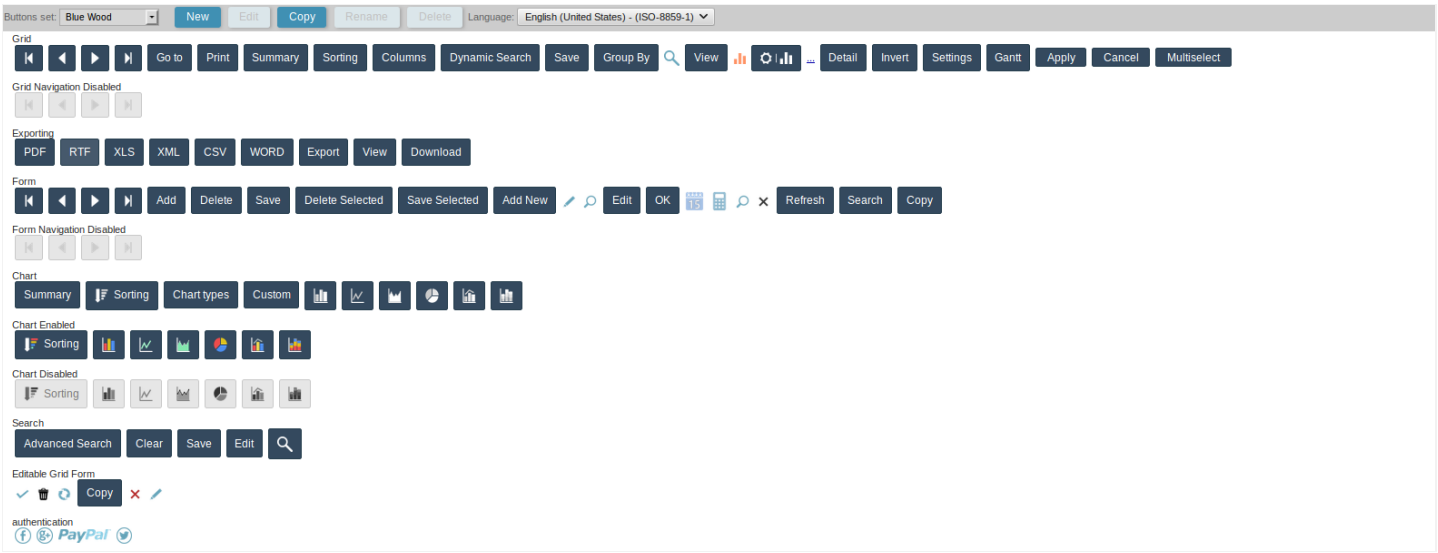
To create new themes, click on "Create New Theme". And the modifications done to the theme, will be displayed on the preview in real-time.



CSS Butons Overview

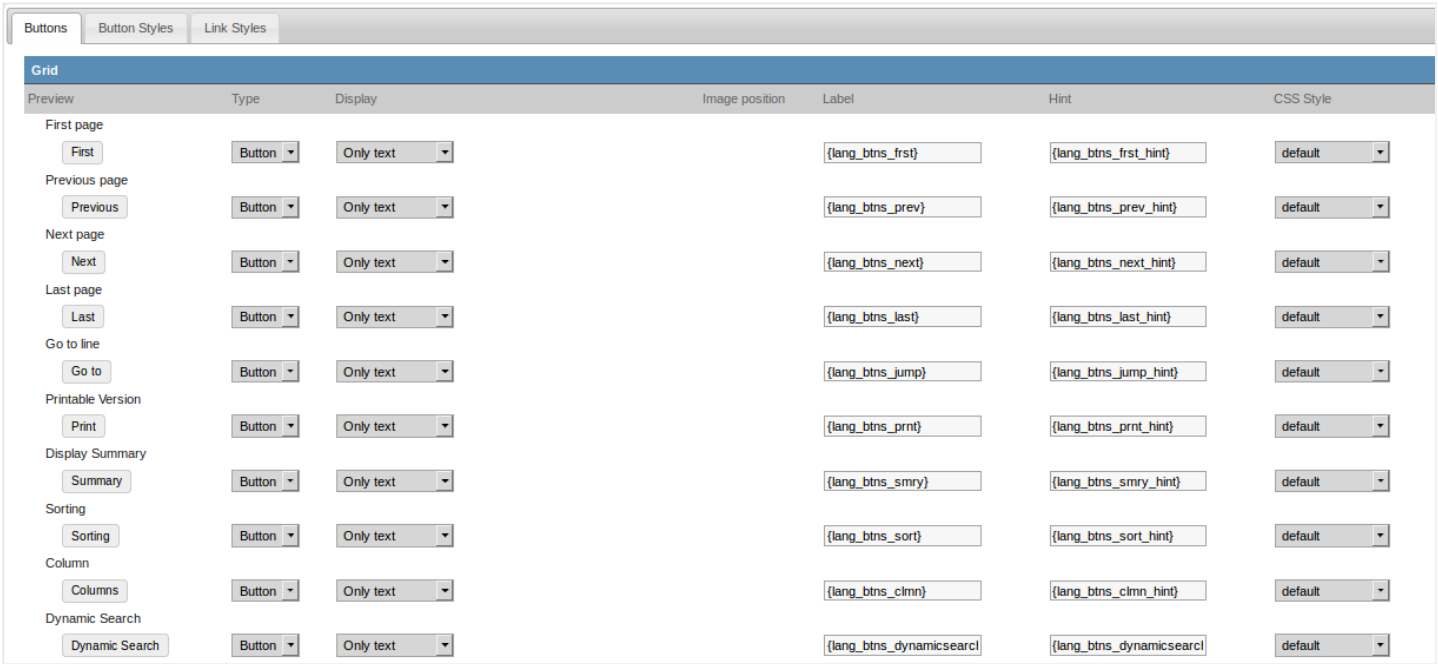
General Overview

In Button Editing we can create new sets of buttons to associate them with themes. We can create or edit sets of text and image buttons.



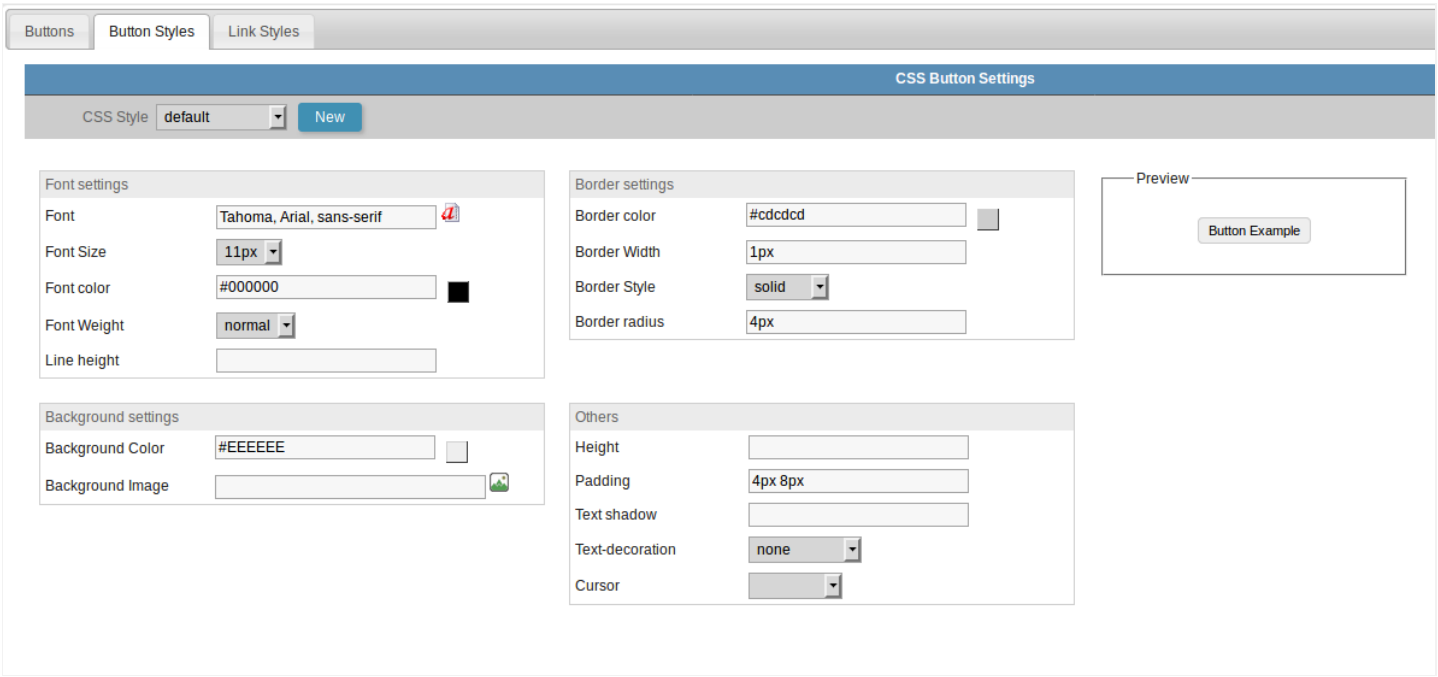
Creating a New Button Scheme

Click the “New” button on the Button Scheme toolbar.



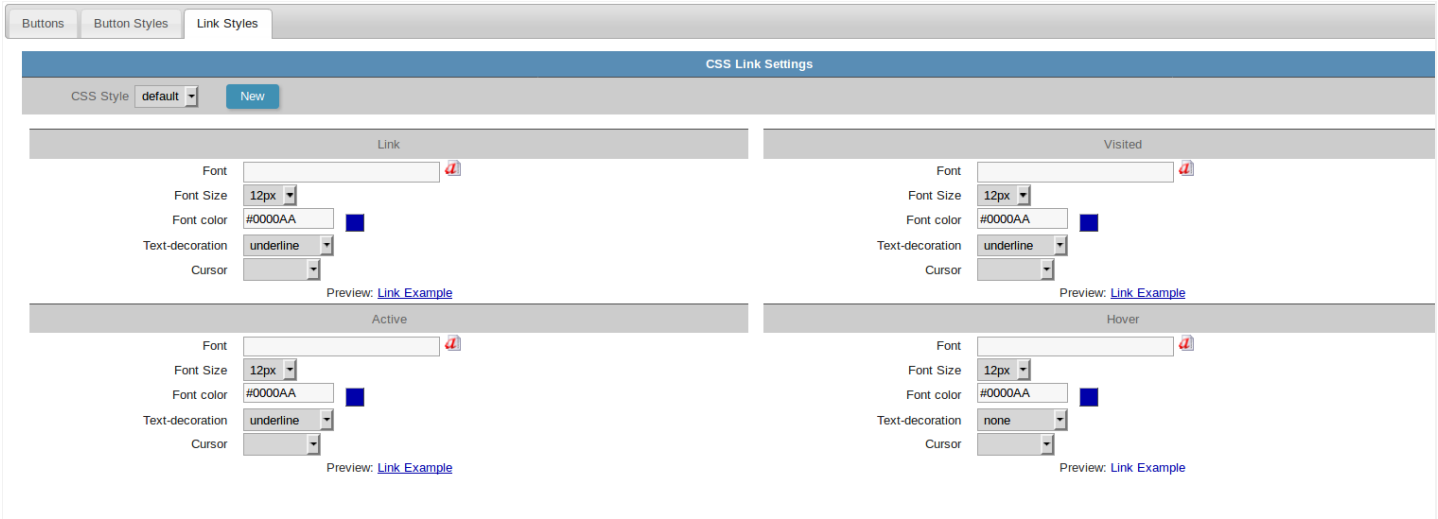
In the first tab (Buttons) the buttons used in Scriptcase are displayed. Each button can have a different characteristic, being possible to choose between image, text button or link. If you use the image type button, a field will be displayed to upload the image. For text and link button types use the tabs: Button Style and Link Styles to edit their appearances.

Editing button styles



In this tab you can define the CSS style of the text buttons used in your outline. All schemes already have a default style that can not be deleted. If your layout requires more than one type of layout for the text buttons, just click on the “new” button and set up a new style. Go back to the Buttons tab and select the created style.

Configuring Link Button Style



Using the same principle exposed in the item styles of text buttons is applied to the styles of links. So you can define that one or more buttons will be of type link and define your CSS style or a new one that you will create.

Related Video ▷

New CSS Button Creation Overview

Creating new buttons set, make possible to edit each component from the Buttons theme so the user can change directly all the sets of buttons, links, and their styles.

Buttons

Here is allowed to configure the default buttons to the applications and components.

See below the available items:

Grid	Authentication
Export Email	Menu
Exporting	Others
Form	Panels And Modal
Chart	Calendar
Chart Enabled	SweetAlert
Search	Editable Grid Form

Button Styles

In this option it is possible to edit the styles that will be used in the buttons set, being available to the user with the following settings:

Button

Button	Button Disabled
Button Hover	Button Selected
Button Onclick	

Dropdown

Container
Line
Button Hover

Link Styles

In this option, it is possible to edit the styles that will be used in the buttons links.

Link Styles

RTL Flag

This option allows that be defined the writing orientation used by the theme. Can be defined if the option **RTL(Right to Left)** will be used.

Grid

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Grid Navigation Disabled

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

CSS Button Settings for Export Emails

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

CSS Button Settings for Export

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Form

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Form Navigations Disabled

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Chart

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Chart Enabled

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Chart Disabled

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Search

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Editable Grid Form

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Authentication

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Menu

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Others

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Modal/App Div

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Calendar

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

SweetAlert

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Button

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Button Hover

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Button Onclick

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Button Disabled

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Button Selected

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Container

Into the Scriptcase buttons settings we can find the main layout settings among them we can find the ‘Dropdown’ and ‘Button’ options.

If what we need is to customize the dropdown style menu then we can select the option ‘Dropdown’ then some interesting options are available:

- Container
- Line
- Button Hover
- Button Disabled
- Button Selected
- Button OnClick

The container is the border of the menu, we can change some settings including:

- Border
- Background
- Others

Border

The border offer us some settings:

Width

This option allows you to define the button border width size.

Color

This option allows you to define the container border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Line

Into the Scriptcase buttons settings we can find the main layout settings among them we can find the ‘Dropdown’ and ‘Button’ options.

If what we need is to customize the dropdown style menu then we can select the option ‘Dropdown’ then some interesting options are available:

- Container
- Line
- Button Hover
- Button Disabled
- Button Selected
- Button OnClick

The container is the border of the menu, we can change some settings including:

- Border
- Background
- Others

Border

The border offer us some settings:

Width

This option allows you to define the button border width size.

Color

This option allows you to define the container border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Button Hover

Into the Scriptcase buttons settings we can find the main layout settings among them we can find the ‘Dropdown’ and ‘Button’ options.

If what we need is to customize the dropdown style menu then we can select the option ‘Dropdown’ then some interesting options are available:

- Container
- Line
- Button Hover
- Button Disabled
- Button Selected
- Button OnClick

The container is the border of the menu, we can change some settings including:

- Border
- Background
- Others

Border

The border offer us some settings:

Width

This option allows you to define the button border width size.

Color

This option allows you to define the container border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Link Styles

CSS Link Settings

Here we can configure the links that will be shown as Buttons in Scriptcase. Handling various visuals aspects related to these links.

Also is allowed to create new CSS Link Styles clicking in the button **New**.

Link

Font

This option allows you to configure the font type that will be used in the link.

Font Size

This option allows you to configure the font size that will be used in the link.

Font Color

This option allows you to configure the link color for the exhibition.

Text-decoration

This option allows you to configure the line personalization type that the link will have.

Cursor

This option allows you to configure the cursor type that will be shown when passing the mouse through the link.

Visited

Font

This option allows you to configure the font type that will be used in the link.

Font Size

This option allows you to configure the font size that will be used in the link.

Font Color

This option allows you to configure the link color for the exhibition.

Text-decoration

This option allows you to configure the line personalization type that the link will have.

Cursor

This option allows you to configure the cursor type that will be shown when passing the mouse through the link.

Active

Font

This option allows you to configure the font type that will be used in the link.

Font Size

This option allows you to configure the font size that will be used in the link.

Font Color

This option allows you to configure the link color for the exhibition.

Text-decoration

This option allows you to configure the line personalization type that the link will have.

Cursor

This option allows you to configure the cursor type that will be shown when passing the mouse through the link.

Hover

Font

This option allows you to configure the font type that will be used in the link.

Font Size

This option allows you to configure the font size that will be used in the link.

Font Color

This option allows you to configure the link color for the exhibition.

Text-decoration

This option allows you to configure the line personalization type that the link will have.

Cursor

This option allows you to configure the cursor type that will be shown when passing the mouse through the link.

Editing CSS Buttons Overview

Creating new buttons set, make possible to edit each component from the Buttons theme so the user can change directly all the sets of buttons, links, and their styles.

Buttons

Here is allowed to configure the default buttons to the applications and components.

See below the available items:

Grid	Authentication
Export Email	Menu
Exporting	Others
Form	Panels And Modals
Chart	Calendar
Chart Enabled	SweetAlert
Search	Editable Grid Form

Button Styles

In this option it is possible to edit the styles that will be used in the buttons set, being available to the user with the following settings:

Button

Button	Button Disabled
Button Hover	Button Selected
Button Onclick	

Dropdown

Container
Line
Button Hover

Link Styles

In this option, it is possible to edit the styles that will be used in the buttons links.

Link Styles

RTL Flag

This option allows that be defined the writing orientation used by the theme. Can be defined if the option **RTL(Right to Left)** will be used.

Related Video ▷

- New buttons and how to customize them

Grid

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Grid Navigation Disabled

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Edit CSS Buttons for Export Emails

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Edit CSS Buttons for Export

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Form

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Form Navigations Disabled

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Chart

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Chart Enabled

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Chart Disabled

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Search

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Editable Grid Form

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Authentication

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Menu

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Others

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Modal/App Div

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Calendar

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

SweetAlert

In this option we can configure the layout of the buttons that will be displayed in the applications used in the project.

Below you can see the list of buttons you can configure.

Type

In this option you can define how the button will appear in the application.

The available options are:

Image

If you select Image, the button appears as an icon in the application.

Button

If you select the button type you can perform the button configuration that will be displayed in the applications.

Link

If you select the Link Type the button in the application will be displayed as a link.

Image

Option Preview

It shows a preview of how the button will appear in the application according to the settings made.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Button

Display

Option to set what will be displayed on the button, the options are:

Only Text

Only the text defined by the user will be displayed.

Only Image

Only the image defined by the user will be displayed.

Text and Image

Both text and images defined by the user will be displayed.

Only Font-awesome icon

The image selected by the user will be displayed.

Display text and icon Font-awesome

Both text and images defined by the user will be displayed.

Icon

Option that allows the upload of own icons to the Scriptcase image manager, in order to be used in the buttons.

Image Position

Sets where the image will be placed on the button, the options are:

Image on the right

The image will be displayed on the right and the text will be displayed on the left.

Text on the right

The text will be displayed on the right and the image will be displayed on the left.

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Link

Label

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

Hint

Option to enter text configured in [Translate Applications][translate_applications] or text entered by the user that will be shown when the mouse is over the button.

CSS Style

Option to set which style will be used by buttons. Layouts are created in the Button Styles or Link Styles options.

Related Video ▷

- New buttons and how to customize them

Button

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Related Video ▷

- New buttons and how to customize them

Button Hover

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Related Video ▷

- New buttons and how to customize them

Button Onclick

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Related Video ▷

- New buttons and how to customize them

Button Disabled

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Related Video ▷

- New buttons and how to customize them

Button Selected

Font

Family

This option allows you to select the family and the font type that will be used in the button.

Color

This option allows you to select the color that will be shown in the button.

Size

This option allows you to select the button font-size in pixels.

Weight

This option allows you to configure the button **font-weight**.

Shadow

This option allows you to configure the shadow of the button font.

Decoration

This option allows you to select an underline type for the button font.

Style

This option allows you to select the style type that will be applied to the button font.

Border

Width

This option allows you to define the button border width size.

Color

This option allows you to define the button border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Height

This option allows you to configure the button height.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Transition

This option allows you to change the button values with a specific duration.

Related Video ▷

- New buttons and how to customize them

Container

Into the Scriptcase buttons settings we can find the main layout settings among them we can find the ‘Dropdown’ and ‘Button’ options.

If what we need is to customize the dropdown style menu then we can select the option ‘Dropdown’ then some interesting options are available:

- Container
- Line
- Button Hover
- Button Disabled
- Button Selected
- Button OnClick

The container is the border of the menu, we can change some settings including:

- Border
- Background
- Others

Border

The border offer us some settings:

Width

This option allows you to define the button border width size.

Color

This option allows you to define the container border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Related Video ▷

- New buttons and how to customize them

Line

Into the Scriptcase buttons settings we can find the main layout settings among them we can find the ‘Dropdown’ and ‘Button’ options.

If what we need is to customize the dropdown style menu then we can select the option ‘Dropdown’ then some interesting options are available:

- Container
- Line
- Button Hover
- Button Disabled
- Button Selected
- Button OnClick

The container is the border of the menu, we can change some settings including:

- Border
- Background
- Others

Border

The border offer us some settings:

Width

This option allows you to define the button border width size.

Color

This option allows you to define the container border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Related Video ▷

- New buttons and how to customize them

Button Hover

Into the Scriptcase buttons settings we can find the main layout settings among them we can find the ‘Dropdown’ and ‘Button’ options.

If what we need is to customize the dropdown style menu then we can select the option ‘Dropdown’ then some interesting options are available:

- Container
- Line
- Button Hover
- Button Disabled
- Button Selected
- Button OnClick

The container is the border of the menu, we can change some settings including:

- Border
- Background
- Others

Border

The border offer us some settings:

Width

This option allows you to define the button border width size.

Color

This option allows you to define the container border color.

Style

This option allows you to select the style type that will be applied to the button border.

Radius

This option allows you to configure the border button radius.

Background

Color

This option allows you to define the button background color.

Image

This option allows you to define a image to the button background.

Repeat

This option allows you to configure the repeating for the image selected to the button.

Position

This option allows you to configure the selected image align for the button.

Shadow

This option allows you to configure the shadow of the button background.

Others

Opacity

This option allows you to configure the button opacity.

Padding

This option allows you to configure the button padding element.

Margin

This option allows you to configure the button margin.

Cursor

This option allows you to select the cursor type that will be shown when you pass the mouse over the button.

Related Video ▷

- New buttons and how to customize them

Link Styles

CSS Link Settings

Here we can configure the links that will be shown as Buttons in Scriptcase. Handling various visuals aspects related to these links.

Also is allowed to create new CSS Link Styles clicking in the button **New**.

Link

Font

This option allows you to configure the font type that will be used in the link.

Font Size

This option allows you to configure the font size that will be used in the link.

Font Color

This option allows you to configure the link color for the exhibition.

Text-decoration

This option allows you to configure the line personalization type that the link will have.

Cursor

This option allows you to configure the cursor type that will be shown when passing the mouse through the link.

Visited

Font

This option allows you to configure the font type that will be used in the link.

Font Size

This option allows you to configure the font size that will be used in the link.

Font Color

This option allows you to configure the link color for the exhibition.

Text-decoration

This option allows you to configure the line personalization type that the link will have.

Cursor

This option allows you to configure the cursor type that will be shown when passing the mouse through the link.

Active

Font

This option allows you to configure the font type that will be used in the link.

Font Size

This option allows you to configure the font size that will be used in the link.

Font Color

This option allows you to configure the link color for the exhibition.

Text-decoration

This option allows you to configure the line personalization type that the link will have.

Cursor

This option allows you to configure the cursor type that will be shown when passing the mouse through the link.

Hover

Font

This option allows you to configure the font type that will be used in the link.

Font Size

This option allows you to configure the font size that will be used in the link.

Font Color

This option allows you to configure the link color for the exhibition.

Text-decoration

This option allows you to configure the line personalization type that the link will have.

Cursor

This option allows you to configure the cursor type that will be shown when passing the mouse through the link.

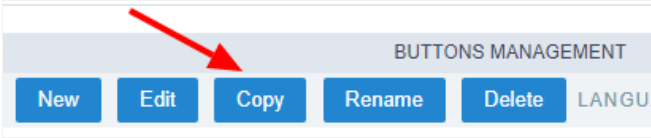
Related Video ▷

- New buttons and how to customize them

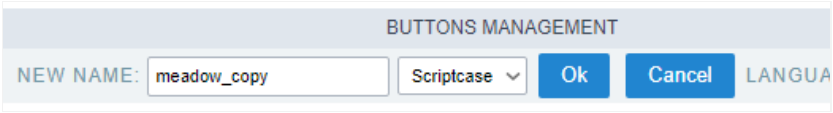
Copy

Copying a Button Set

To copy a Button set, it is necessary to click on the button **“Copy”** available between the buttons CSS settings toolbar.



After that, will be displayed the copy settings options to the buttons, see below:



New Name

This field allows the user to define the buttons set copy name.

Set Level

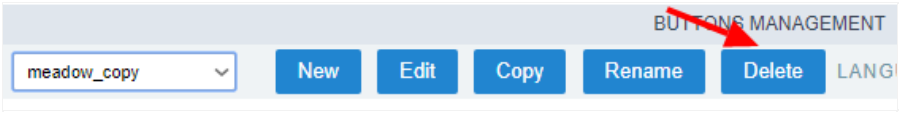
This option allows selecting the level that the buttons set will have. Can choose between: **Scriptcase, Public, Project and User**.

Related Video ▷

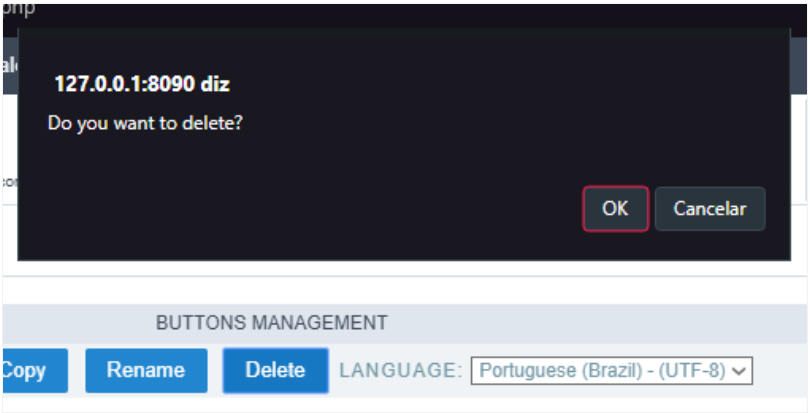
Delete

Deleting a Buttons Set

To delete a specific button set, you need just to select the desired set and click on the delete button, as the image:



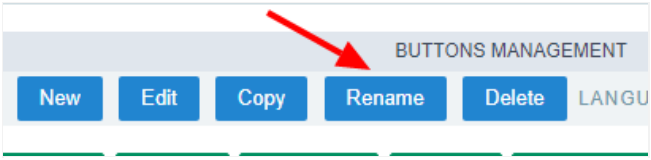
After that, will be displayed a confirmation pop-up for deleting, you will need just click on the button **OK** and the set will be deleted.



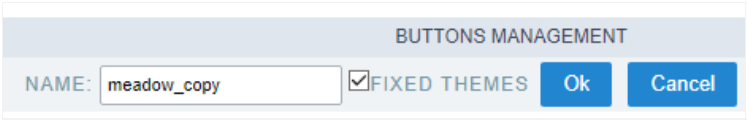
Rename

Renaming a Buttons Set

To rename a specific button set, you will need just select the desired set and click on the button**Rename**, as the image:



After that, will be displayed some options, so you can rename the set.



Name

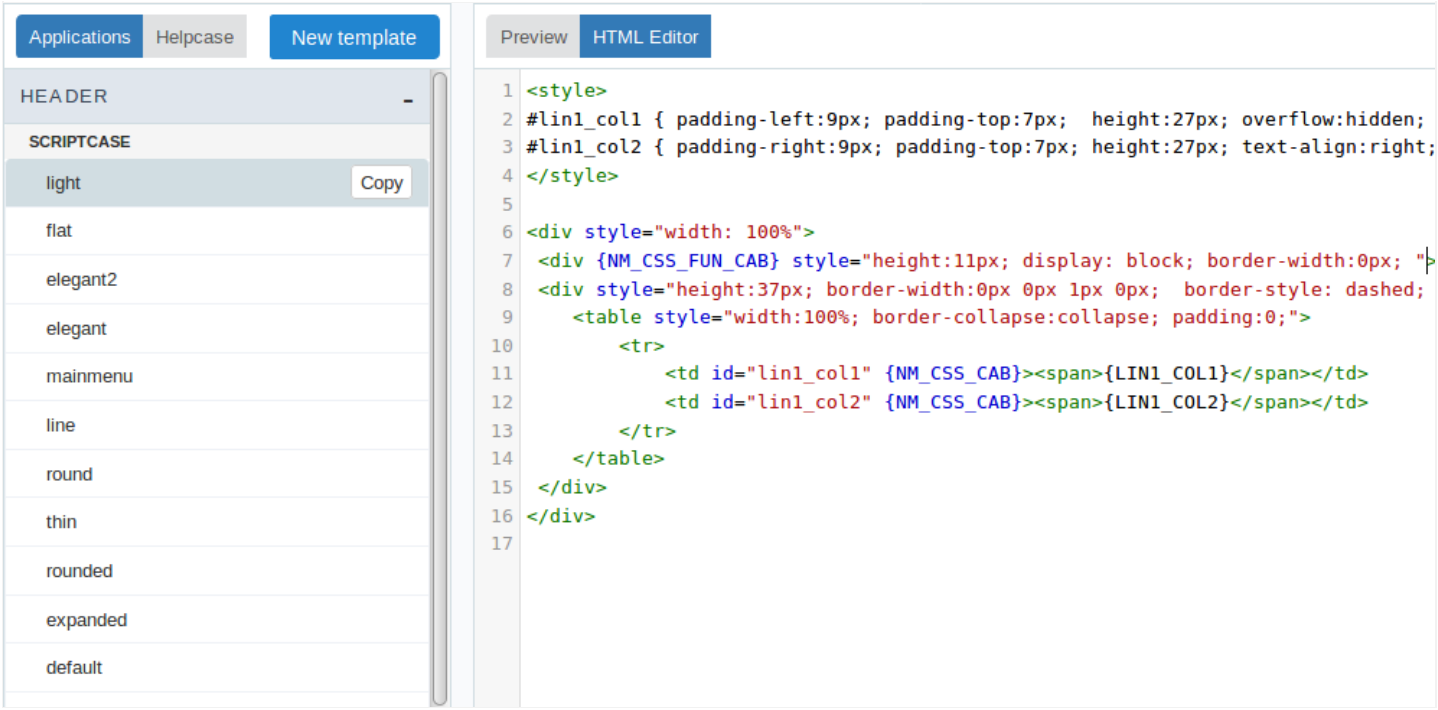
This field allows you to inform the name that will be used in the renamed set.

Fixed Themes

Marked as default, this option allows that when changes the set name, this change will be applied in the projects that use the button set.

HTML Templates

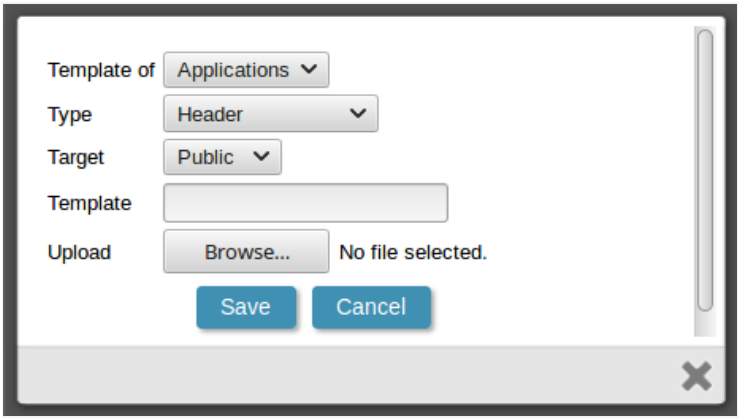
In the Templates, it's possible to edit and create models for the Headers, Footers or the User HTML (Grid and Control). These models define the interface structure of the applications. You can create a HTML file and define where the entry data will be, by use curly brackets {} to specify a variable. The User Defined HTML option is used in Grid and Control applications, where it's possible to define all the structure of the report in HTML and associate the fields of the applications with the variables created on the body of the HTML file. With this it's possible to create Control and Grid applications with a different layout, being able to place the fields anywhere on the page.



Header and Footer

In the editor for the Header and Footer, you need to create or edit existing models.

To create a new model, there's a standard code to use as a base of all the HTML.



The creation of fields in the HTML needs to be done with curly brackets, for example: {variable}.

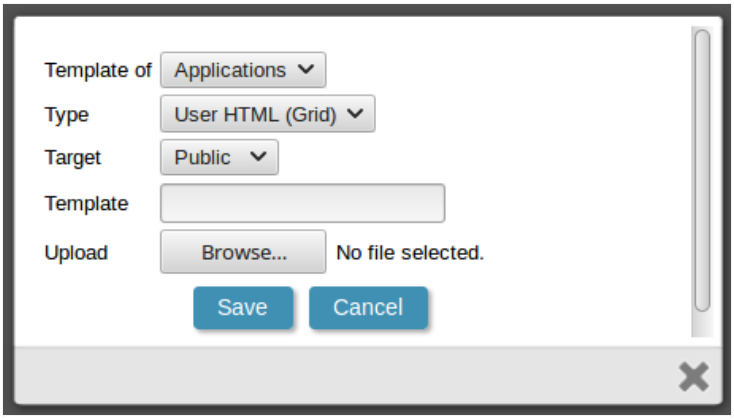
Code example:

```
<div style="width: 100%">
 <div {NM_CSS_FUN_CAB} style="height:11px; display: block; border-width:0px; "></div>
 <div style="height:37px; background-color:#FFFFFF; border-width:0px 0px 1px 0px;  border-style: dashed; border-color:#ddd; c
   <table style="width:100%; border-collapse:collapse; padding:0;">
      <tr>
         <td id="lin1_col1" {NM_CSS_CAB}><span>{LIN1_COL1}</span></td>
         <td id="lin1_col2" {NM_CSS_CAB}><span>{LIN1_COL2}</span></td>
      </tr>
   </table>
 </div>
</div>
```

These fields are displayed in the application where the template is going to be used.

After saving the template, you need to open the application that you're going to use the template in and go to "Layout > Settings", choose the model for the Header and Footer that you'll use and in "Layout > Header & Footer" you'll associate the created fields in the template with the desired information available by ScriptCase.

User Defined HTML



In the User Defined HTML we can specify where the fields of the applications will be displayed in the HTML file by using curly brackets {variable}. Like the example below:

```
<table border="0" cellspacing="0" cellpadding="0">
    <tr>
          <td width="247" height="96" align="left" valign="center">
          <font face="verdana" style="font-size:11px">
          <b>{Name}</b><br>{Address}<br>{City} - {State}<br>
          {ZIP}
          </font>
          </td>
    </tr>
</table>
```

The preview of the code above will be like the following: {Name} {Address} {City}-{State} {ZIP}

Remember that all the content that has curly brackets are considered a field of the application, this is why when you have {Name}, it'll be referencing a field called 'Name' in your application.

In the template we can define a delimiter by using "":

```
<table border="0" cellspacing="0" cellpadding="0">
    <tr>
        <!-- BEGIN bl2 -->
            <td width="247" height="96" align="left" valign="center">
            <font face="verdana" style="font-size:11px">
            <b>{Name}</b><br>{Address}<br>{City}- {State}<br>
            {ZIP}
            </font>
            </td>
        <!-- END bl2 -->
    </tr>
</table>
```

These parts are defined where the loop is going to start and end. The quantity of delimiters are defined as the “Columns per page” on the Grid application.

e.g.: If within the Grid application the option “Lines Per Page” is set to 3, you’ll see the following result.

```
<table border="0" cellspacing="0" cellpadding="0">
    <tr>
            <td width="247" height="96" align="left" valign="center">
            <font face="verdana" style="font-size:11px">
            <b>{Name}</b><br>{Address}<br>{City}- {State}<br>
            {ZIP}
            </font>
            </td>
            <td width="247" height="96" align="left" valign="center">
            <font face="verdana" style="font-size:11px">
            <b>{Name}</b><br>{Address}<br>{City}- {State}<br>
            {ZIP}
            </font>
            </td>
            <td width="247" height="96" align="left" valign="center">
            <font face="verdana" style="font-size:11px">
            <b>{Name}</b><br>{Address}<br>{City}- {State}<br>
            {ZIP}
            </font>
            </td>
    </tr>
</table>
```

The preview of the code above will be the following:

{Name}	{Name}	{Name}
{Address}	{Address}	{Address}
{City}-{State}	{City}-{State}	{City}-{State}
{ZIP}	{ZIP}	{ZIP}

If you have a HTML File created in the standards above, you can include it in ScriptCase by accessing “Layout > HTML Templates > User HTML > New Template”. On “New Template” you’ll inform the file that you want to import or create a new template by writing the code in the ScriptCase editor.

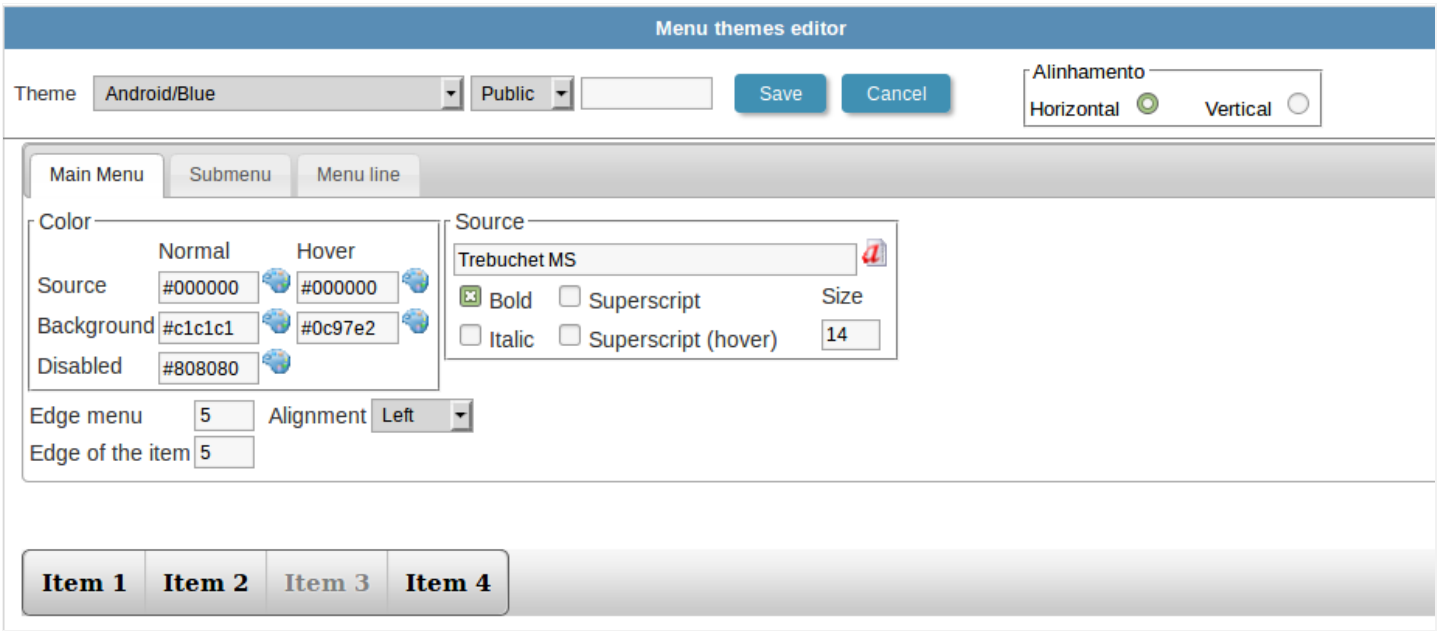
After having the User Defined HTML, the next step is open the application that supports the Template (Control or Grid) and go to “Position of Fields” and associate the fields with the application fields.

CSS Menu

The CSS Menu section is used to create and/or edit layout styles to be applied in Menu Applications.

To create a new theme for the menu it's necessary to first select a theme to be used as base of the new theme.

After clicking on "New", to create a new theme for the Menu, you'll have many options to change the colors that are part of the Menu you chosen. Also, you can change the font size of the items.



Before saving the theme that you created, you need to select the access level as Public, Project or User levels.

- **Public** the theme is available for everyone that uses your Scriptcase installation.
- **Project** the theme is available for the project that it was created in.
- **User** the theme is available only for the user that created the theme.

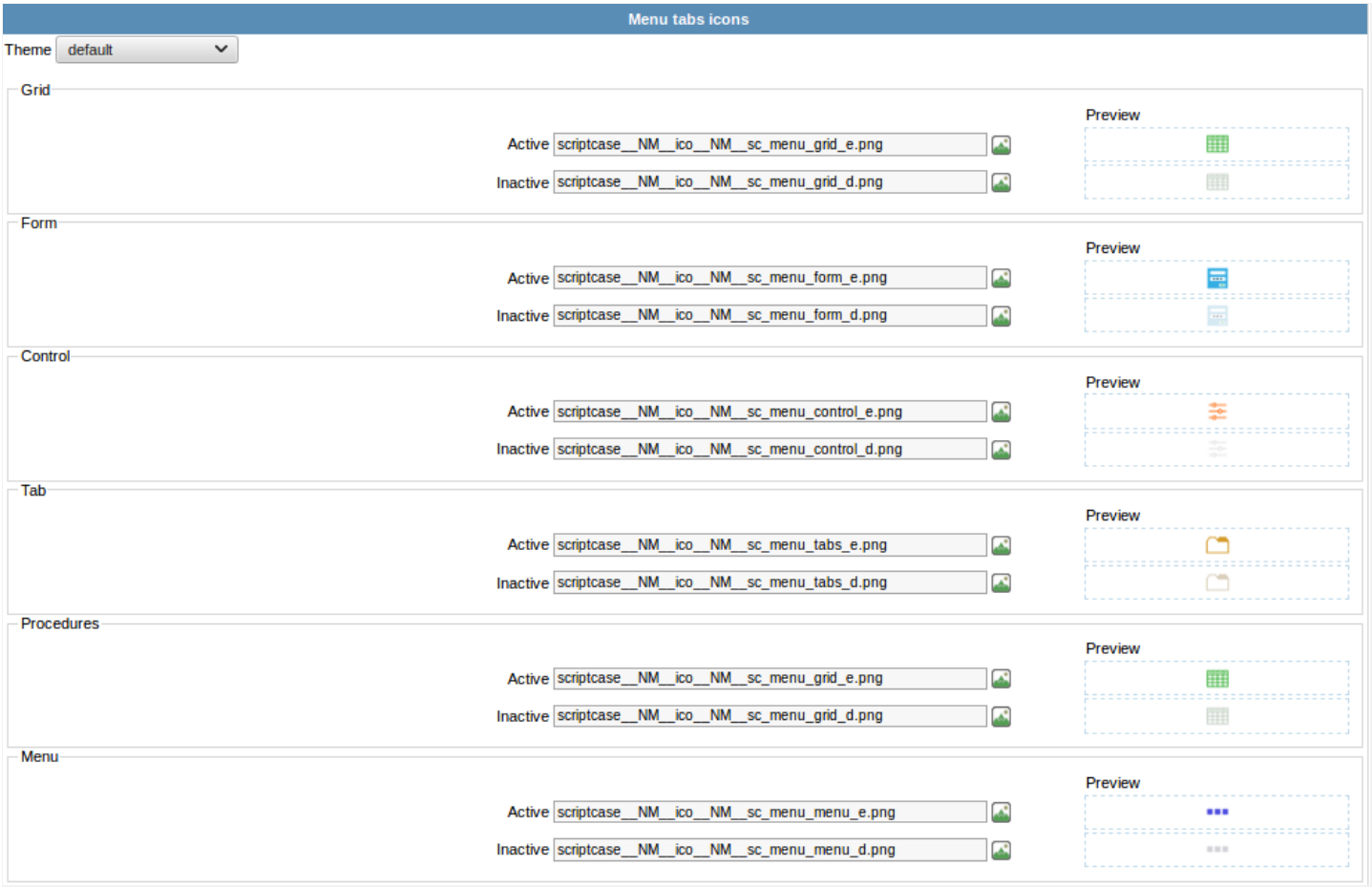
After selecting the access level, inform the name of the theme and click on "Save".

Menu Icons

While editing the menu icons, you can change the icons of the active and inactive tabs that will be opened in the menu. These icons are linked to the opened applications.

On the page below, we can view the icons divided in blocks for each type of application, so each icon corresponds to an application from ScriptCase.

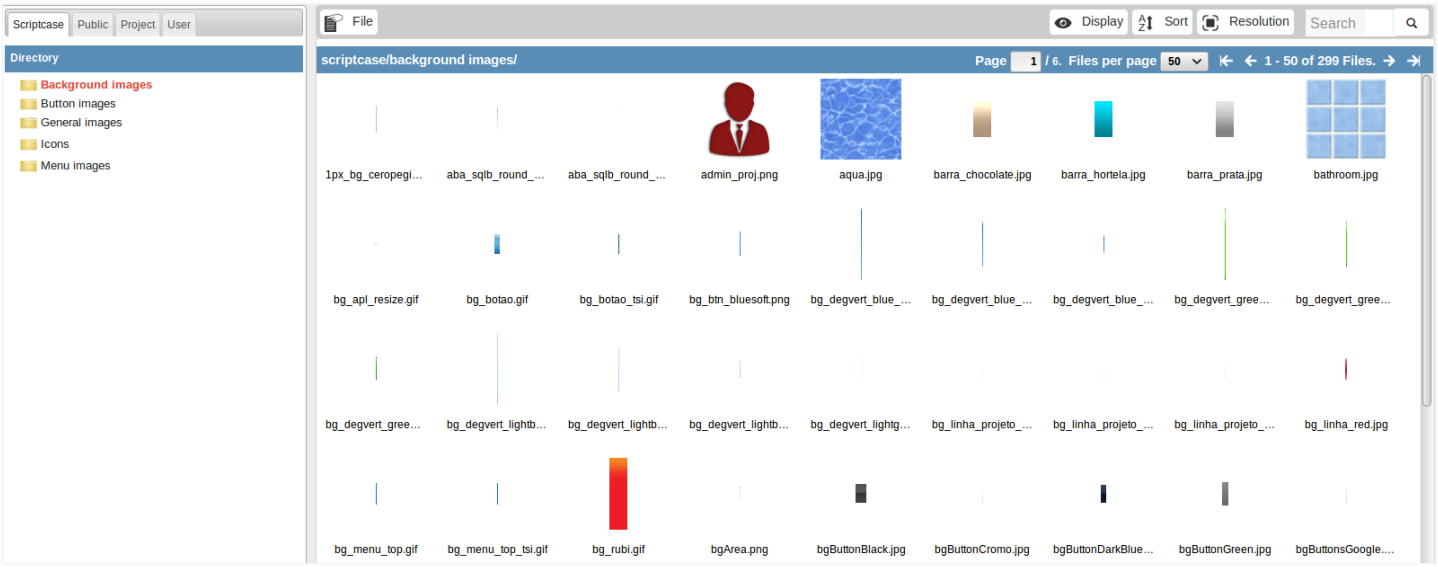
Select the icons that you desire for each state for the menu tabs, being active or inactive.



Images Manager

The Images Manager is where you'll organize and upload new images that'll be used in your project.

To upload new images, you need to select the scope of which the image will be part, after selecting the folder where the image will be stored at and after that click on File > Upload.



When the image is uploaded, the image's name is modified depending on the scope of which it's part of (public, project or user) and the type of image (background, button, icons, menu or general) following the pattern below:

- sys__NM__ (Public)
- grp__NM__ (Project)
- usr__NM__ (User)

Image Type:

- bg__NM__ (background images)
- btn__NM__ (button images)
- img__NM__ (general images)
- ico__NM__ (icons)
- menu_img__NM__ (menu images)

This way, the image called 001.jpg that was added to the public and icon scope will receive the name sys__NM__ico__NM__001.jpg

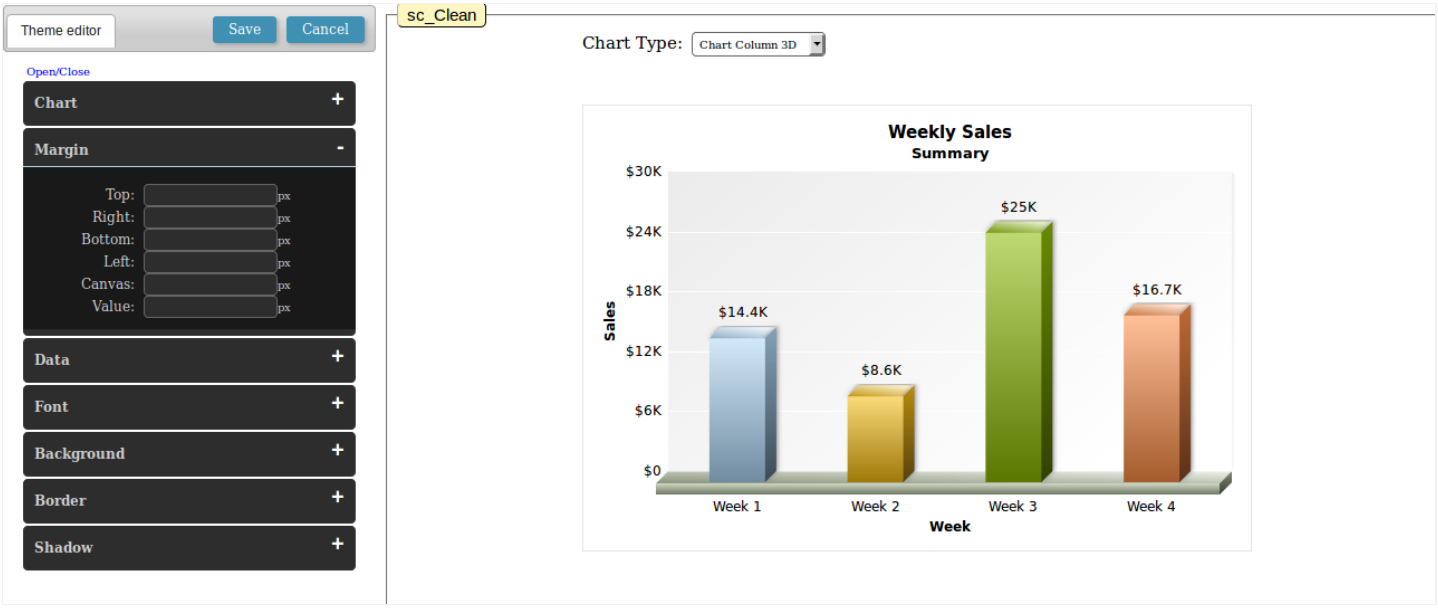
The image will be in the folder "app/project_name/_lib/img" in the development environment and in the production, will be "/project_name/_lib/img".

After uploading the image to the project, you need to insert the image in the applications that you desire, to do that in the applications desired, go to Applications > Settings > Application images and click on the icon next to the garbage icon to add an image.

Chart Themes

The chart themes tool offers the option for you to create different themes for the chart applications of your project. The chart theme allows you to modify some options, such as the background color, font, margin, data, border and shadow.

To create new themes, you need to select an existing theme so that you can use it as a base and on top of it you'll customize it to the way you feel satisfied.

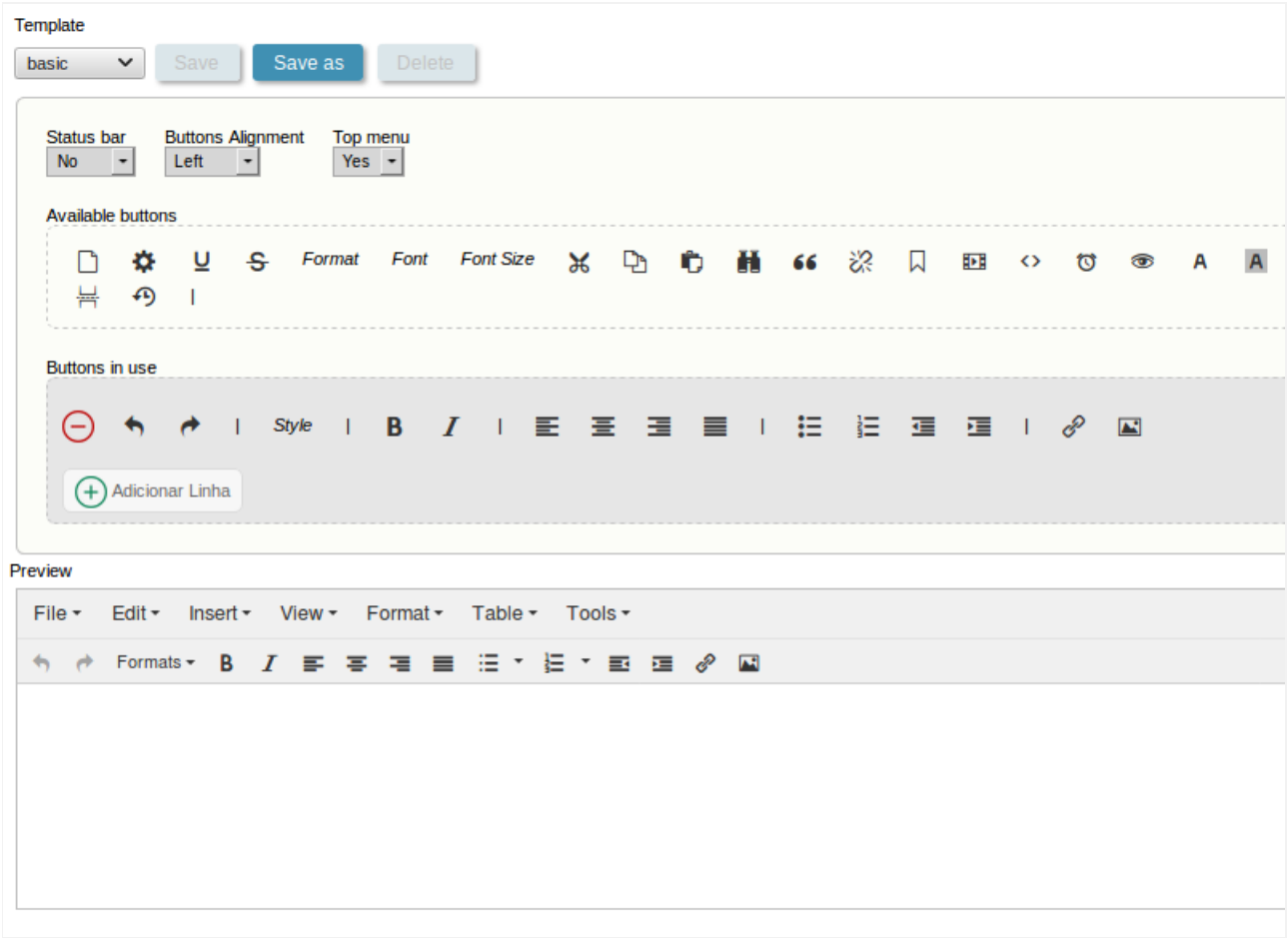


HTML Editor Templates

The HTML editor tool offers you the option to modify the toolbar of the html editor field present in the applications.

You can create a new template theme and/or edit an existing one.

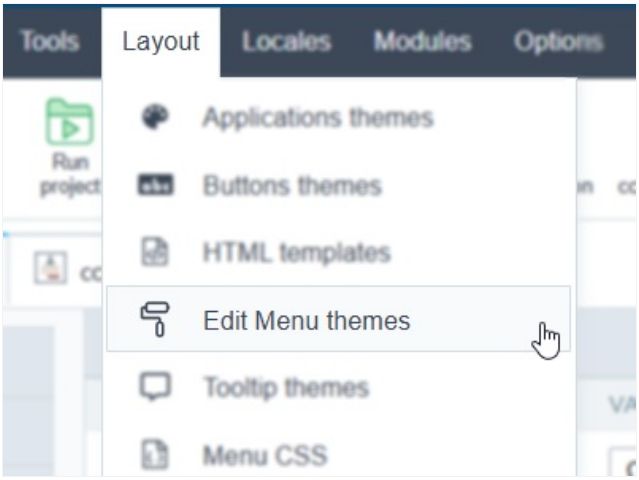
To create a new theme, just select an existing one and click on “Save As”, to edit an existing theme just select the theme and click on “Save”.



Edit Menu themes

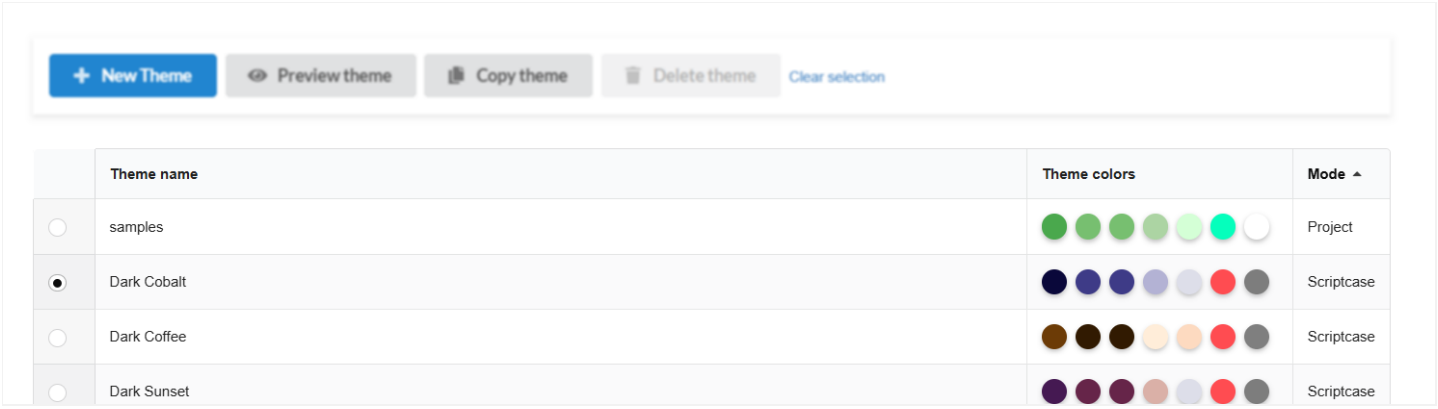
The theme editor allows developers to easily create new themes for the responsive menu, changing the application's color palette to match the system's visual design.

To access the theme editor, go to **Layout > Edit Menu Themes**.



Theme List

When accessing the editor, the existing themes in Scriptcase are displayed, sorted by the saving mode (scope).

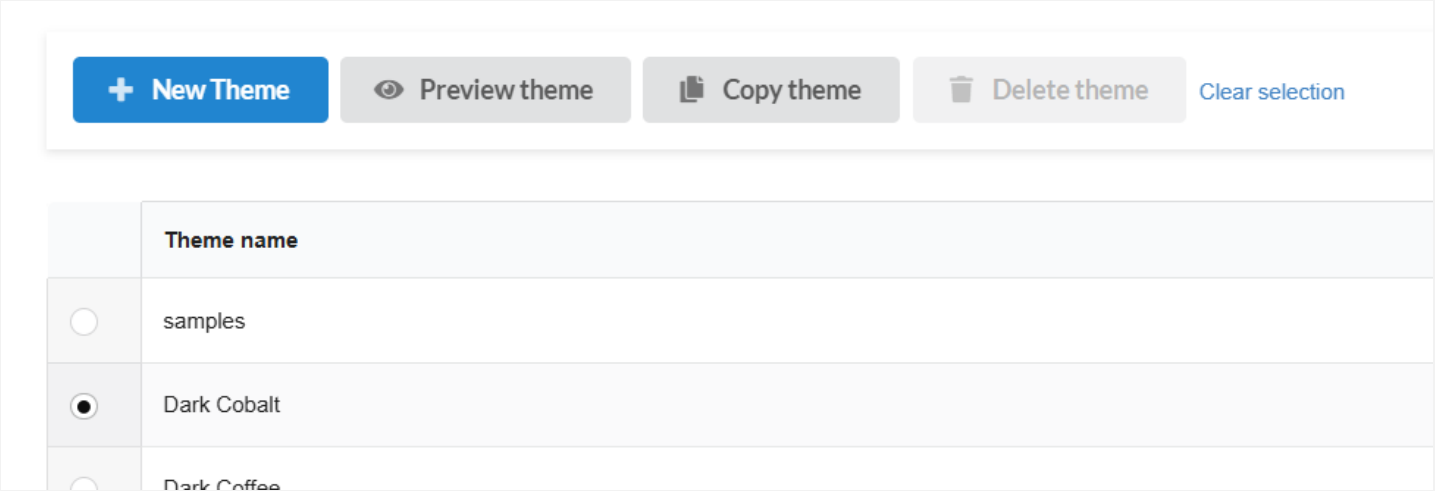


- **Theme name** - Displays the theme names.
- **Theme colors** - Displays the colors that make up the theme.
- **Mode** - Indicates the scope in which the theme was saved.
  - **Scriptcase**: Default themes provided by Scriptcase. These themes cannot be edited or deleted but can be copied to create new themes.
  - **Public**: Themes available for all projects.
  - **Project**: Themes available only for the current project.
- 🗑 - Removes the selected theme. This icon is displayed only for themes created by the developer.

Creating Themes

Menu themes can be created from scratch by clicking on **New Theme**, or based on one of the themes provided by Scriptcase, by selecting the desired theme and clicking **Copy Theme**.

Scriptcase provides some themes that can serve as a foundation for creating new ones.



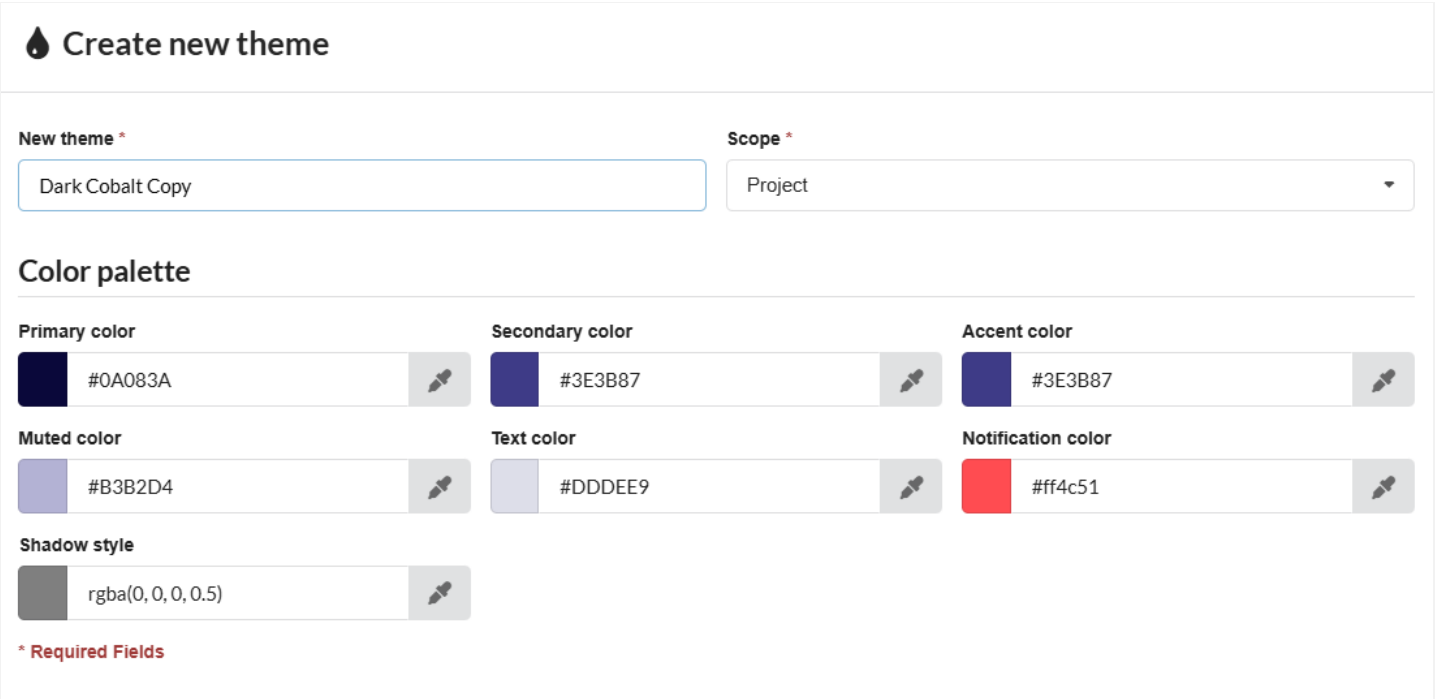
- **New theme** - Opens the theme creation screen with no predefined colors.
- **Preview theme** - Allows previewing the selected theme.
- **Copy theme** - Creates a copy of the selected theme if it is one provided by Scriptcase.
- **Clear selection** - Deselects the selected theme.

The **Copy theme** button changes to **Edit theme** when selecting a theme created by the developer.

Copying a Theme (Copy Theme Button)

The **Copy theme** button is enabled when selecting one of the themes provided by the tool. After selecting and clicking the button, the **Create new theme** screen will open with the color palette filled in from the base theme.

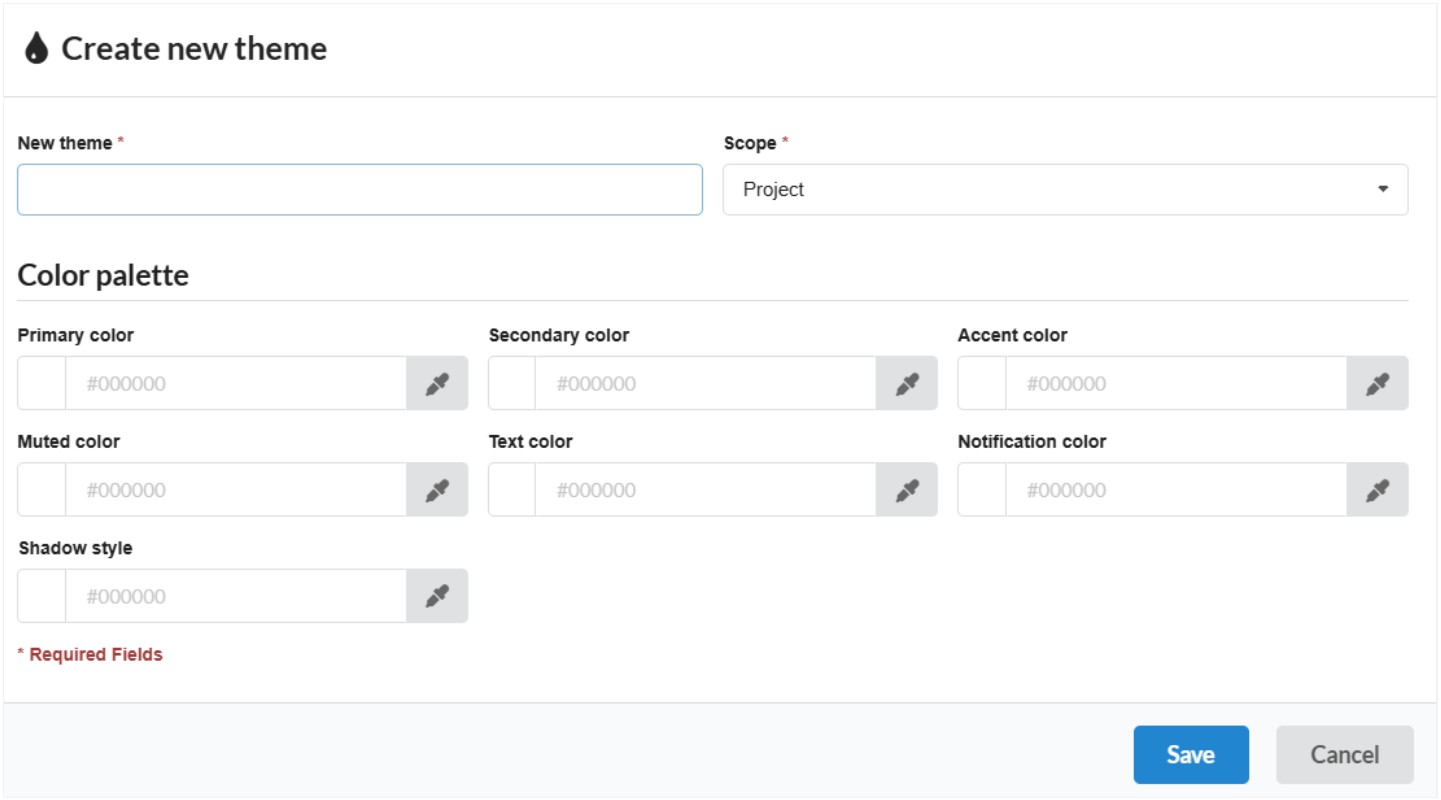
New Theme Screen (Copy Theme Button)



Blank Theme (New Theme Button)

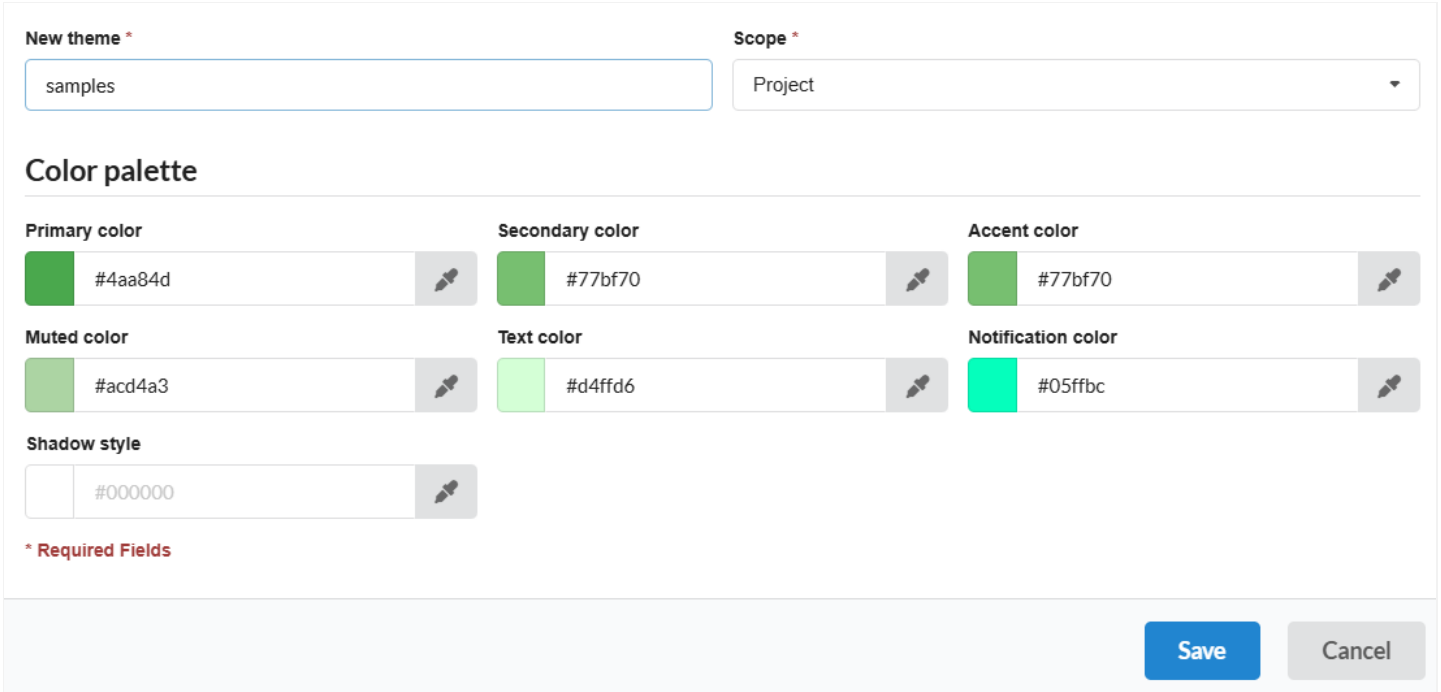
After clicking the **New theme** button, the **Create new theme** screen will open with no predefined values.

New Theme Screen (New Theme Button)



Theme Editor

The theme editing screen is displayed for creating a new theme or editing existing themes.

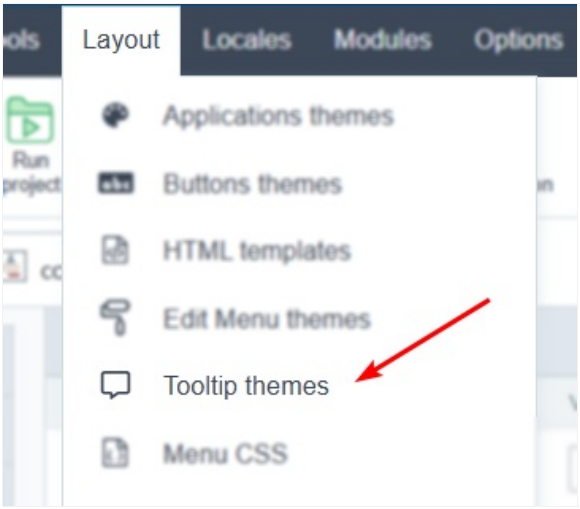


- **New theme** - Defines the name of the theme.
- **Scope** - Defines the scope in which the theme will be saved.
- **Color palette** - Defines the theme's color palette.

Tooltip themes

This theme editing tool allows developers to modify the default TippyJS themes provided by Scriptcase. Creating a custom CSS theme for TippyJS lets you adjust the appearance of tooltips to match the visual style you want to apply, offering greater control and customization of the interface.

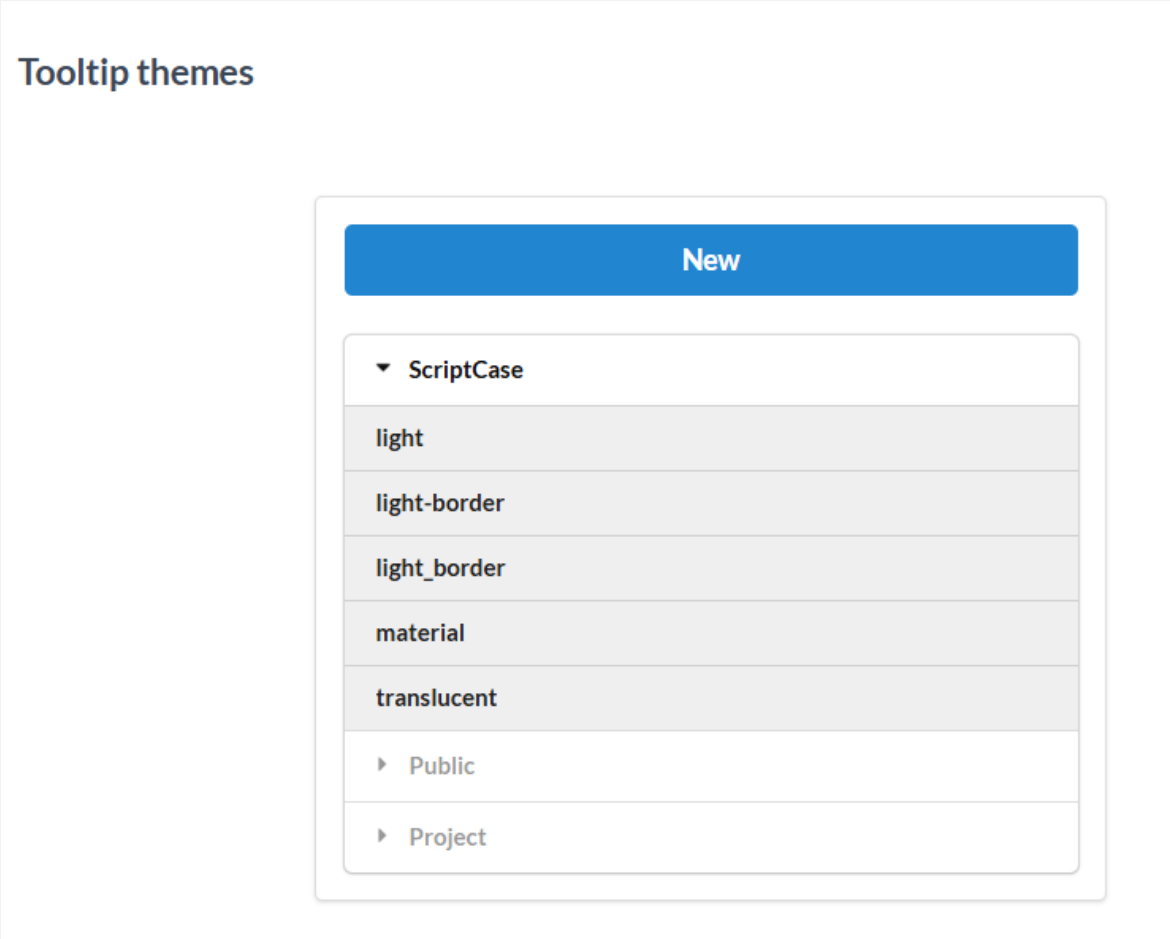
To access the theme editor, go to **Layout > Tooltip Themes**.



Theme List

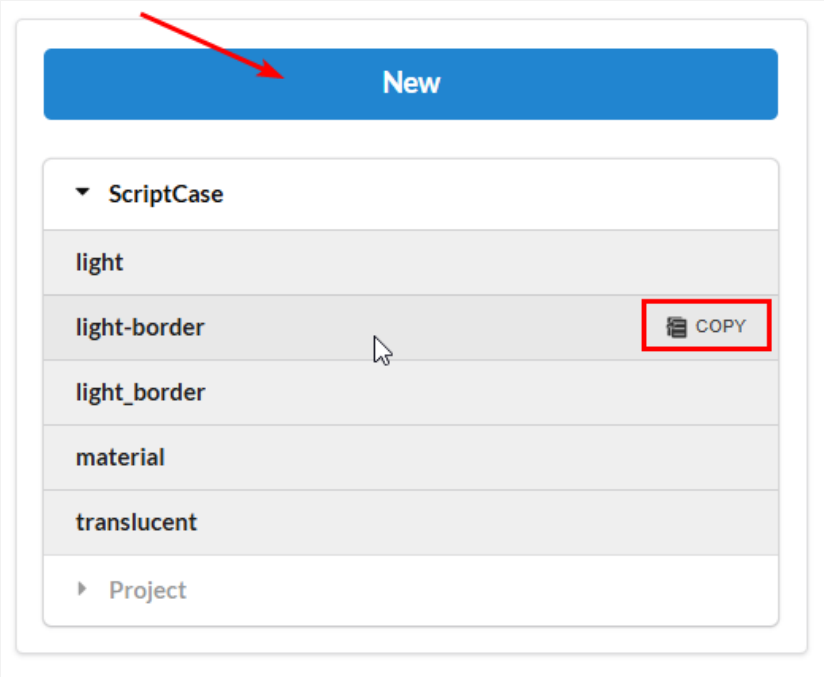
When accessing the theme editor, the existing themes will be listed, grouped according to their scope.

- **Scriptcase**: Default themes provided by Scriptcase. These themes cannot be edited or deleted but can be copied to create new ones.
- **Public**: Themes available for all projects.
- **Project**: Themes available only for the current project.



Creating Themes

New TippyJS themes can be created from existing Scriptcase themes or from scratch by clicking the **New** button.



Copying Themes (Copy Button)

The themes provided by Scriptcase can serve as a foundation for creating new themes. By clicking the Copy button, which

appears when hovering over a specific theme, the developer can create a copy to use as a base for CSS customization.

Blank Theme (New Button)

By clicking the **New** button, the editing screen will open with a blank theme, allowing the developer to create a theme from scratch.

There is no specific structure required to create a TippyJS theme. However, maintaining a basic structure already used by the component can simplify the application of styles and ensure that TippyJS correctly recognizes the custom theme.

Below is a basic template for a TippyJS theme.

```
/* Basic structure of a custom TippyJS theme */

/* Tooltip box */
.tippy-box[data-theme~="theme-name"] {
  /* Tooltip background color */
  background-color: #f0f0f0;

  /* Text color inside the tooltip */
  color: #333;

  /* Tooltip rounded corners */
  border-radius: 6px;

  /* Tooltip shadow */
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);

  /* Font and text size */
  font-size: 14px;

  /* Tooltip padding */
  padding: 10px;
}

/* Tooltip content (the text or HTML inside it) */
.tippy-box[data-theme~="theme-name"] .tippy-content {
  /* Sets content padding */
  padding: 8px 12px;

  /* Font styling for the content */
  font-family: Arial, sans-serif;
}

/* Arrow customization */
.tippy-box[data-theme~="theme-name"] .tippy-arrow {
  /* Arrow size */
  width: 12px;
  height: 6px;
}

/* Arrow when the tooltip is above the target (top) */
.tippy-box[data-theme~="theme-name"][data-placement^="top"] > .tippy-arrow::before {
  /* Arrow color to match the tooltip background */
  border-top-color: #f0f0f0;
}

/* Arrow when the tooltip is below the target (bottom) */
.tippy-box[data-theme~="theme-name"][data-placement^="bottom"] > .tippy-arrow::before {
  border-bottom-color: #f0f0f0;
}

/* Arrow when the tooltip is to the left of the target (left) */
```

```
/* Arrow when the tooltip is to the left of the target (left) */
.tippy-box[data-theme~="theme-name"][data-placement^="left"] > .tippy-arrow::before {
  border-left-color: #f0f0f0;
}

/* Arrow when the tooltip is to the right of the target (right) */
.tippy-box[data-theme~="theme-name"][data-placement^="right"] > .tippy-arrow::before {
  border-right-color: #f0f0f0;
}

/* Transition effect (appearance and disappearance animation) */
.tippy-box[data-theme~="theme-name"] {
  /* Smooth opacity transition when showing/hiding the tooltip */
  transition: opacity 0.3s ease-in-out, transform 0.3s ease-in-out;
}

/* Tooltip when it is visible */
.tippy-box[data-theme~="theme-name"][data-state="visible"] {
  /* Tooltip is fully visible */
  opacity: 1;
  transform: scale(1);
}

/* Tooltip when it is hidden */
.tippy-box[data-theme~="theme-name"][data-state="hidden"] {
  /* Tooltip is partially invisible while hidden */
  opacity: 0;
  transform: scale(0.95); /* "Shrink" effect when disappearing */
}
```

Code Editor

CSS editing screen for creating a TippyJS theme

- **Mode** - Defines the scope in which the theme will be saved. The developer can choose between:
  - **Project** - The created theme will be available only in the current project.
  - **Public** - The created theme will be available in any project.
- **Name** - Sets the name for the new theme. By default, when copying a theme, *_copy* is added after the original theme's name.
- **Theme** - Defines the CSS code editor's theme.
- 🔍 - Enables the code editor's search feature.
- 🔄 - Enables the code editor's replace feature.

Scriptcase Locales Overview

ScriptCase offers a complete generator so that you can define the languages and regional settings that your project will support. The locales settings can be accessed via the menu Locales.

Application Language

The language menu for the applications allows you to translate the ScriptCase's default messages, and also create custom messages to be used in different parts of the applications.

Too see how to set the Application Language, click here.

Regional Settings

The regional settings allows you to define some parameters of the monetary unit, date and numbers, according to the country or region where your application is going be used. Clicking on Customizing, you can modify the values that'll be used in your applications.

To see how to set the Regional Settings, click here.

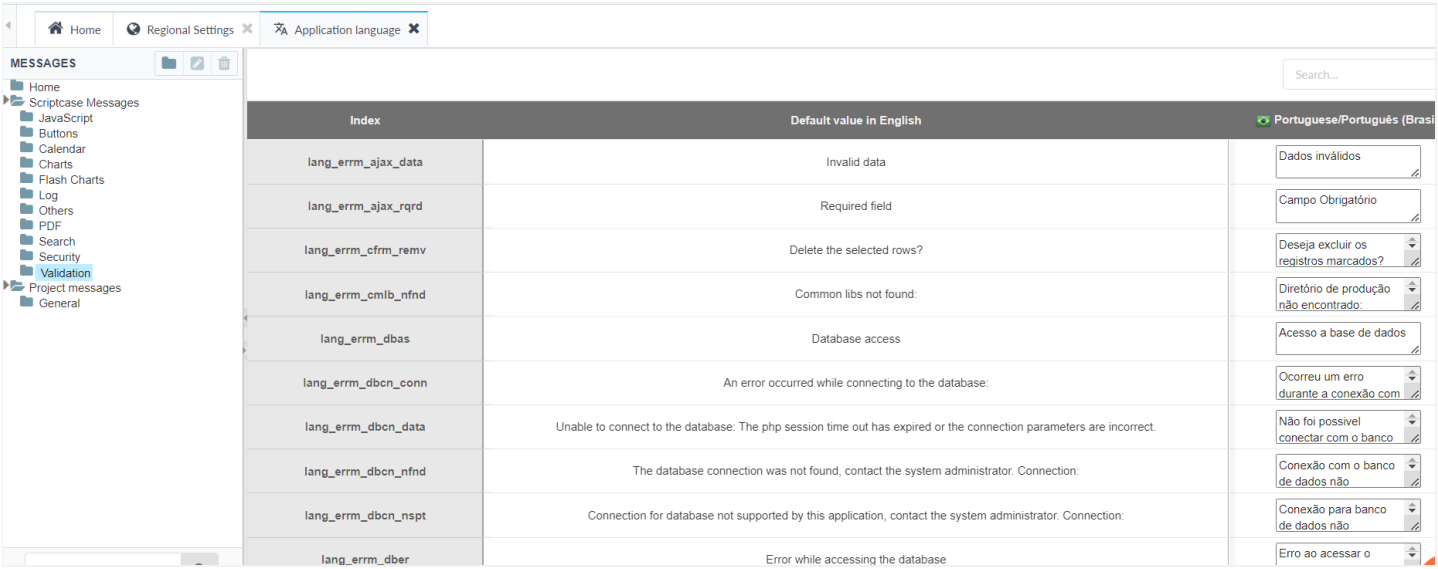
Application Language

The language menu for the applications allows you to translate the ScriptCases default messages, and also create custom messages to be used in different parts of the applications.

Messages

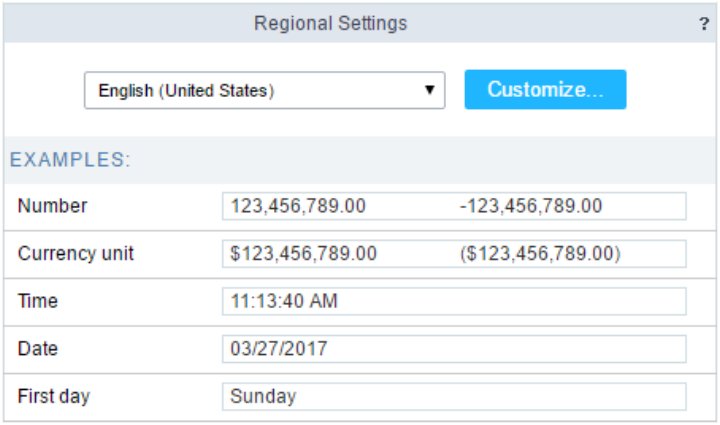
Using the messages menu, it's possible to select and edit messages. The messages are divided in the categories: "ScriptCase Messages" and "Project Messages". New categories can be created using the item "New Folder".

Selecting a lang to be modified, a new page will appear with the columns of each language of your project. You'll inform the value of the message in each language.



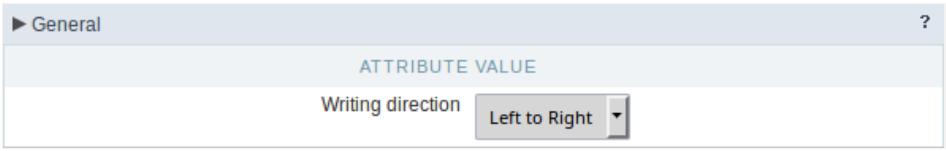
Regional Settings

The regional settings allows you to define some parameters of the monetary unit, date and number according to the country or region where your application is going be used. Clicking on Customize you can modify the values that'll be used in your applications.

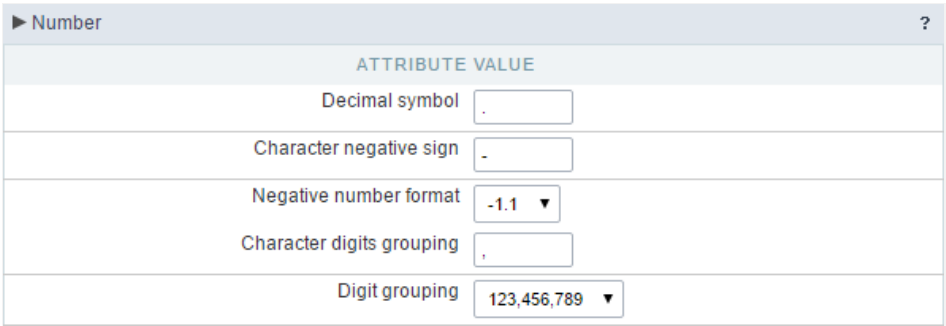


Customizing Regional Settings

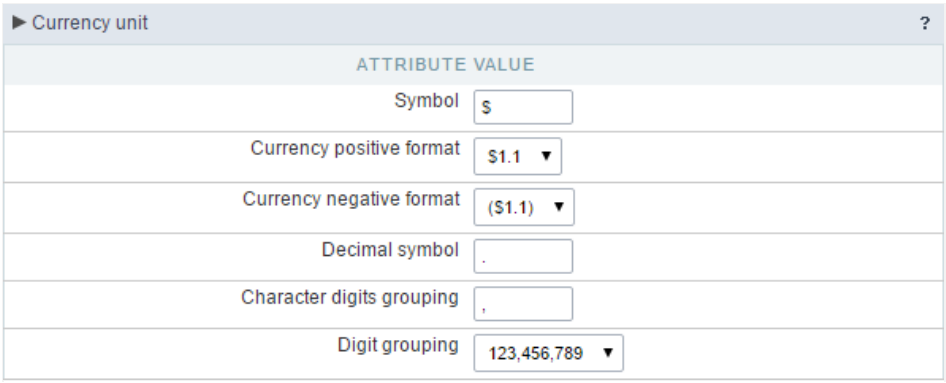
In the General group, you can set the writing direction. This setting configures if the characters will be inserted from left-to-right or right-to-left.



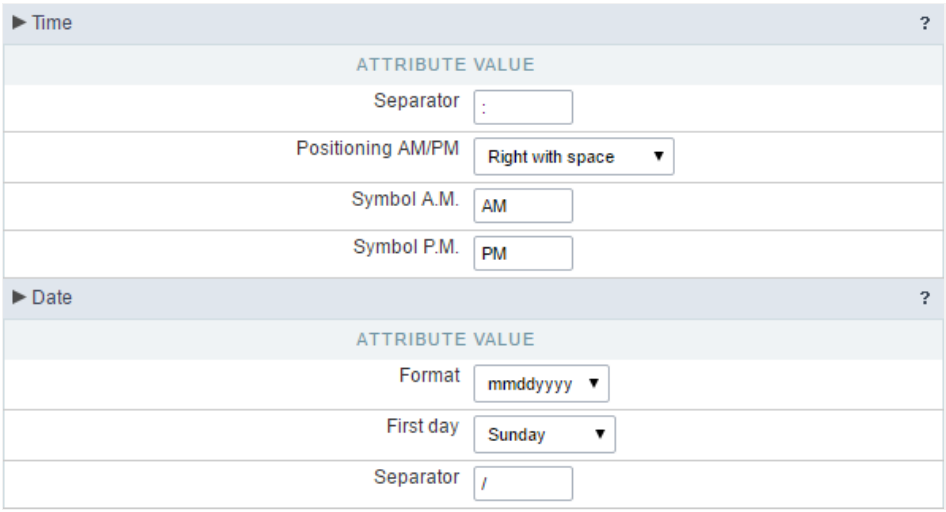
In the Number group, we have some parameters like the Decimal Symbol, digit grouping and negative format. These parameters are used in the applications that uses the decimal type field.



In the Currency Unit group, we have some parameters like the Monetary Symbol, Currency positive format, etc. These parameters are used in the applications that uses the Currency type field.



Within the Time and Date groups are available the parameters of Date and time, that can be used in the fields of the Type Time, Date and Datetime.



Modules Overview

Scriptcase offers a comprehensive set of security features, integrating security and logging modules to quickly establish efficient access control across your project.

Old Security Module

This module provides system protection, creating access control and defining privileges for users, ensuring that only authorized users have access to certain applications.

Five types of security modules are available, which work in a similar way, their main distinction is in the way permissions are applied to users.

Click here to access documentation about the Old Security Module.

New Security Module

Its operation is similar to the traditional security module, however, it has more configuration options and makes all these options available in the **Configuration** application, allowing the system's security settings to be changed at the end, by the system administrator, making the security module more dynamic.

Click here to access documentation about the New Security Module.

Notifications Module

The Notification Module automatically generates a set of applications used in the management of system notifications. This module allows you to create access control with restrictions according to the type of Notification module.

Click here to access the documentation about the notifications module.

Log Module

Records events and activities in the system, creating a detailed history of actions such as login, data updates and other operations. Logs are essential for monitoring user changes to the system and are valuable in audits.

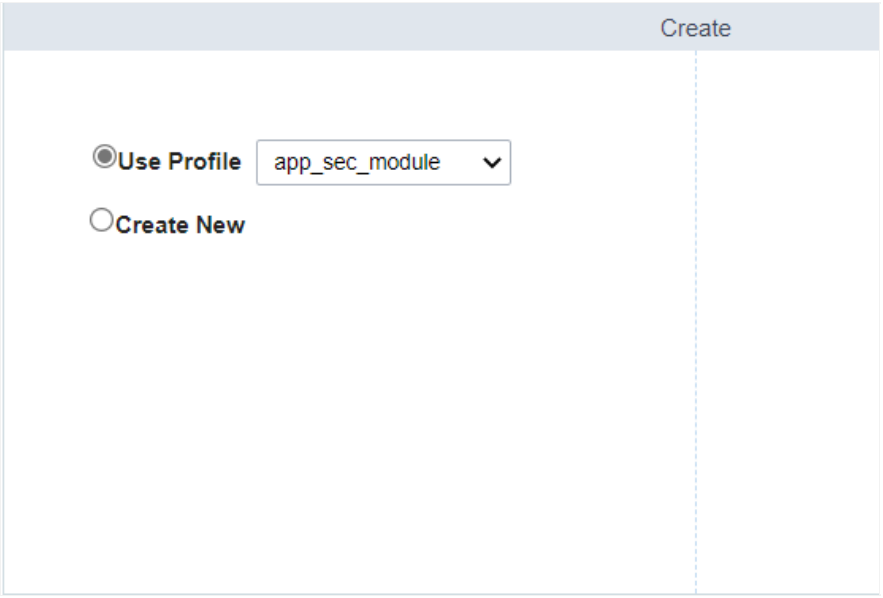
Click here to access documentation about the logging module.

Security Module Overview

Create

This screen is displayed when accessing the security module if there is any previously configured security profile.

On this screen, the developer chooses to create a new security module or use an existing profile.



Use Profile

When selecting this option the module configurations are filled according to the configurations used in the selected profile creation.

The creation of profiles in the security module occurs at the end of the module's configuration process, storing the options and settings made.

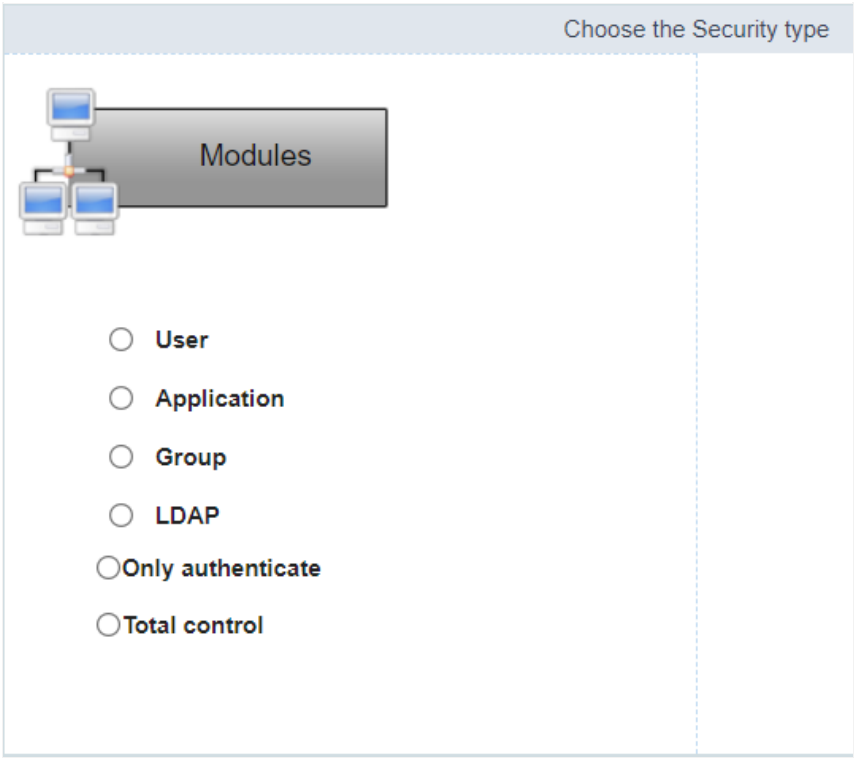
Profiles can be saved in three ways.

- **Public** - With this option all projects and users of the tool will have access to this profile.
- **Project** - When saving the profile at the project level, all users of the project where the profile was created will have access for use..
- **User** - Only the user who performed the profile configuration will have access to its use in any project.

Create New

When choosing to create a new module, the module's configurations change to Scriptcase's default values.

Choose the Security Type



This is the first step in creating the security module.

On this screen, we must select the type of security that will be created. In this case the **User module** will be chosen.

Security module per User

Under this security type all users have access to all applications. It works only for user authentication. Scriptcase will create one table to store the user and password information and validate the access using a login screen.

For more details about the **User Security Module**, click here

Security module per Application

Under this type Scriptcase will also control login and password as the User Security and also create a restricted access control to the applications or system options according to the logged user. It is possible to define which applications is accessible to each user.

For more details about the **Application Security Module**, click here

Security module per Group

This type of security includes the options of User and Application types, however with the Group Security type you will group the users and define the permissions to access the applications or system options according to the groups. One user can belong to one or more groups.

For more details about the **Group Security Module**, click here

Security module per LDAP

The Lightweight Directory Access Protocol (LDAP) is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network. You can use this protocol within Scriptcase Security module to authentication users in two ways:

Authentication only

Performs only a simple user authentication, similar to User Security Type.

For more details about the **Ldap Authentication only**, click here

Total Control

where we can define the access by groups, similar to the Group Security Type.

For more details about the **Ldap Total Control**, click here

Per-User Security Module Overview

Making use of the security modules, you can implement a complete access rules for systems developed by ScriptCase. The process for the security module creation is quick and simple.

Security module per User

In user security, all registered users have access to system applications, working only as user authentication.

For those who want to create the tables in advance, only the user table is mandatory. The logged and users_social tables are created only when activating the Protect Logged users and Use social networks options, respectively.

Below see all the tables created for this type of security.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    PRIMARY KEY ("login")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    "login" TEXT NOT NULL,
    "date_login" TEXT,
    "sc_session" TEXT,
    "ip" TEXT
);
```

> The table **Logged users** will be created only if the option **Protect Logged users** is checked during the Security Module creation.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> The table **users_social** will be created only if the option **Use social networks** is checked during the Security Module creation.

sec_settings

```
CREATE TABLE "sec_settings" (
set_name TEXT NOT NULL,
set_value TEXT,
PRIMARY KEY ("set_name")
)
```

Use Existing Tables

Connection

Define the connection where the security module tables fixed. All existing connections in the project will be displayed

> If you need to create a new connection, click here and see how.

Use existing tables

Allows you to use existing tables in your database. These tables must have, at least, the same fields used by the Scriptcase security module.

> We recommend that you use this option if you have used Scriptcase to create the tables before, to minimize errors.

If you want to know the specifications for this security module, click here.

Create tables

Once this option is selected, Scriptcase is responsible for creating the tables necessary for the use of the security module.

Tables prefix

Defines a prefix for the tables that will be created by the security module.

> By default, the Scriptcase uses "**sec**_".

Delete if tables already exist

This option deletes tables with the same name from your database.

E.G.: When defining the **prj**_ prefix for your tables, if there is any table with the name **prj_users** in your database, it will be deleted.

> This option is only available when selecting a **Create tables** option.

Protect Logged Users

This option prevents the same user from accessing the system simultaneously in different session.

By checking this option, the **logged** table will be created.

> When using the default prefix or the table name will be **sec_logged**.

Use Social Networks

This option allows the configuration of Facebook and Twitter for the authentication of system users.

By checking this option, the **users_social** table will be created.

> When using Social Networks, the **Login Template** option will not be available to define the security module login layout.

Association tables

This step is essential if the option **“Use existing tables”** is selected.

In this case, you must associate the fields from the existing tables to fields of security applications (applications generated by the Security Module).

> If you have selected the option **“Create tables”**, in the previous step, it ssociates the fields automatically.

If you choose to create the tables beforehand, see below as local tables and fields.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    PRIMARY KEY ("login")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    "login" TEXT NOT NULL,
    "date_login" TEXT,
    "sc_session" TEXT,
    "ip" TEXT
);
```

> The table **Logged users** will be created only if the option **Protect Logged users** is checked during the Security Module creation.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> The table **users_social** will be created only if the option **Use social networks** is checked during the Security Module creation.

General

Applications Prefix

You can set a prefix to the application names of the Security Module.

When session expires

Define the behaviour when the session expires.

- **No action** - The user continues using the application, but no saves after the session expires.
- **Redirect to login after the session expiration** - The application returns to the login after expiring the session.
- **Display a message that the session has expired** - Shows the message “session has expired” to the user.

Encryption

Use encryption to store the password in the table of users.

Enable Security

Activate the flag Application Security for all project applications.

Remember login

Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.

Cookie expiration time

Cookie expiration time in days for **Remember login** option

Label Position at Login

Defines the Label positioning of the fields in relation to the data.

- **Beside** - Default value of the applications, placing the label on the right side of the input label to the Side
- **Above** - Positions the label above the input label Above
- **Below** - Positions the label below the input label Below
- **Watermark** - Positions the label as Watermark.

Enable Captcha

It activates the captcha for the login application.

- **No** - Does not display the captcha in the login application.
- **Captcha** - Uses the built-in scriptcase library for captcha display.
- **reCAPTCHA** - Uses Google’s reCAPTCHA V2. To configure, click here.

  Watch a video of reCAPTCHA. If you prefer, see a tutorial on how to generate the keys.

Site Key

Key generated by Google after reCAPTCHA project creation.

Secret Key

Key generated by Google after reCAPTCHA project creation.

Folder

The project folder name to store the applications generated by Security Module.

Theme

The theme to create the Security Module applications.

Log

This option is available if the project already has a Log Module.

> Click here and check out how to create a Log Module

Menu

This option is only available if your project already has a Menu Application. You can associate the existing Menu and include all applications generated by the Security Module to it. If you do not select an existing Menu here, it creates a new Menu Application.

Menu Type

Security Module menu type (option only available if you do not select an existing menu in the previous item)

> This option is available when using the **Create Application Menu** option in the **menu** item

Use SweetAlert

Enables the use of sweetAlert in security module applications.

Settings

Defines whether the security module editing application will be generated.

With this option active, the system administrator user will be able to change some settings of the security module at the end, when accessing the system.

See below for the list of options available for editing the security module.

- **Action for session expiration** - Define the system behavior when the session expires. This feature is triggered as soon as a new action is performed.
  - **No action:** No action will be taken and the user will continue using the system, however no action will be saved.
  - **Redirect to login:** When performing some action on the system, the user will be redirected to the login application.
  - **Displays a message and redirects to login:** or perform some action on the system, a message will be displayed informing you that the session has expired with an ok button. By clicking on the button, the user will be redirected to the login application.
- **Remember login** - Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.
- **Remember login** expiration time - Cookie expiration time in days for the **Remember login** option. By default, Scriptcase sets the limit to 30 days.
- **Recover Password** - Enables the reset password button on the login screen.
- **New Users** - Enables the add new users button on the login screen.
- **Brute force** - Enables or disables the failed access attempt limit.
- **Block time for Brute Force protection** - Defines the time, in minutes, that the user will remain unreachable after several failed login attempts.
- **Brute Force Attempts for blocking** - Defines the number of failed access attempts until protection is activated.
- **2F** - Defines whether two-factor authentication will be used on the system.
- **2FA Expiration Time** - Determines the time in seconds that the 2FA authentication token will expire

Login

Here you can set the type and amount of characters allowed for the username and password fields.

User

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

Password

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

View password characters

This option toggles in the password field, allowing the displayed password to be displayed.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

E-mail Settings

Email Server

Select an email API to use in the security module.

This email will be used in the options:

- All options of **Retrieve Password**
- **New User** when select the options: **Requires activation by email**, **Send email to administrator**.

Custon - Enter the SMTP data in the options below.

> The options below are only available when the **custon** option is selected.

SMTP Server

Enter the SMTP server address.

SMTP Port

Enter the SMTP server port. This information must comply with the secure connection option. Use 465 for SSL, 587 for TLS, or 25 for unsafe connection. If you do not inform the port, Scriptcase applies the default one: 25.

Secure Connection

Use SSL or TSL, or leave it blank for insecure connection.

- SSL
- TSL

SMTP User

Enter the SMTP User information.

SMTP Password

Enter the SMTP password information.

SMTP E-mail

Enter the SMTP outgoing email.

SMTP

> To proceed with the password recovery, it is necessary that has the SMTP server configured for a successful process. Look here how to realize this procedure.

Yahoo

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.
> See also: Ways to securely access Yahoo Mail

SMTP Server Settings Yahoo	
SMTP Server	smtp.mail.yahoo.com
Port	465 or 587
E-mail address	Your full email address (name@domain.com.)

SMTP Server Settings Yahoo	
Password	The password for your account.
Requires SSL	Yes
Requires TSL	Yes (if available)
Requires authentication	Yes

Gmail

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.

SMTP Server Settings Gmail	
SMTP Server	smtp.gmail.com
E-mail address	Your full email address (Ex. example@gmail.com)
Password	The password for your account.
Port (TLS)	587
Port (SSL)	465
TLS / SSL Required	Yes

Outlook / Hotmail

For more information about these settings Click Here

SMTP Server Settings Outlook.com	
SMTP Server	smtp-mail.outlook.com
E-mail address	Your full email address (Ex. myname@outlook.com, not an alias)
Password	The password for your account.
Port	587 (you can use port 25 as an alternative)
TLS / SSL Required	Yes

Retrieve Password

It defines the password recovery method in the project.

> The password recovery options work only when the SMTP server is configured, please check the Email settings.

Send the password by e-mail

The system emails the user password just if the SMTP has been correctly set up, and the password is not using encryption.

Reset password and send new one by email

The system resets the password automatically and sends it to the user by email (just if the SMTP has been correctly set up).

Send a link to e-mail with reset option

The system emails the user a link to access an application and reset the password.

New users

Settings for creating new users in the security system.

> The options activation by email and send an email to admin work only when the SMTP server is configured properly, access Email settings to set it up.

Allows the registration of new users

This option sets the availability of users registration directly in the login system. If you do not check this option, only users with administrative access can insert new users.

Requires activation by email

This option sets whether the new user must validate his registration by email to activate the account (Configure the email SMTP to use this option)

Send email to administrator

This option sets whether the system administrator receives an email whenever a new user is registered. (Configure the email SMTP to use this option)

Social Network

Allows the use of **Facebook** and **Twitter** to login to the security module.

> When using social network authentication the **Login Template** option will not be available.

Facebook

Enable login via facebook. For the login to work, it is necessary to configure an application to obtain the **App ID** and **Secret**

See how to generate facebook credentials

App ID

Enter the **Application ID** available in the application settings on facebook developers.

Secret

Enter the **Application Secret Key** available in the application settings on facebook developers.

Twitter

Enable login via Twitter. For the login to work, it is necessary to configure an application to obtain the **App ID** and **Secret**

App ID

Enter the **Application ID**.

Secret

Enter the **Application Secret Key**

Logged Users

Defines the system behavior for login protection.

This option is available if you have checked Protect logged users during the connection step.

Display logged users

If you check this option Scriptcase will also create with the Security Module a Grid Application to display a report with all users current logged on the system

Brute Force Attack Protection

Enables/disables the blocking of users after some unsuccessful access attempts.

Brute Force lockout time (in Minutes)

Time, in minutes, that the user will remain inaccessible after several unsuccessful access attempts. (Available only when rute Force Attack Protection is enabled)

Numbers of attempts before lock

Number of failed access attempts, until the protection is enabled. (Available only when enable protection for brute force attacks)

Login Template

Allows the use of a custom HTML when creating the security module login application.

This HTML must be previously loaded into an external library, which in turn must be enabled for use in the project. See how to enable an external library.

> This option is only available when **Authentication by social network** is not being used.

Use the template for the lofin screen

Defines whether the login application will use a custom HTML or a standard Scriptcase theme.

No

By checking no, the login application will be created with the project's default theme.

Yes

When checking yes, the **Login Template** button will be enabled, thus allowing the selection of one of the listed templates.

> It is necessary to enable the external library that will be used.

- **Enable external Libraries** - Redirects to external library management screen.
- **Reload** - Reloads the select, listing the unlisted libraries after enabled.

Two Factor Authentication

Authentication (2FA)

In this option the user will select the type of 2 factor authentication to be used, there are three options:

Auth

This option will send the code to your Authentication app from Google.

> See how to use the authentication API

SMS

This option will send the authentication code by SMS.

> See how to use the SMS API

E-mail

This option will send the authentication code by SMS.

> See how to use the email sending APIs

API

In this option you select the API previously created to be used to send the code.

Code expiration time

Time in following to expire token sent for authentication.

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the security system and add new users.

> ATTENTION: To increase the security of your project, we recommend changing the default password “admin/admin” at this step or after the first access.

Login

This option sets the system administrator user.

> By default the defined user is **admin**

Password

This option sets the system administrator password.

> By default the defined password is **admin**

Name

This option sets the name of the administrator user on the system.

E-mail

This option sets the system administrator user E-mail.

> For sending email to the administrator to work, either SMTP is configured correctly or an email API is selected in the Email Settings screen in the previous step.

Save Profile

Using this option, you save all settings during the Security Module creation. It can be used later for other projects.e

Save Profile

Allows you to save a profile with all the current security module settings.

Name

Profile name. It identifies the profile.

Target

This option sets what developers can use the saved profile afterward.

- Public - Set the security profile available in any project of your Scriptcase.
- Project - Set the security profile available only in the current project.
- User - Set the security profile available only to the current Scriptcase user

Building Security

Security module creation summary screen, listing changes made by the tool.

Open Project

Go to the Home of the currently open project.

Generate Source Code

Generates the source code of all applications created by the security module.

> After the creation of the security module, it will be necessary to generate the phone of all the applications, which all had a change in the use security flag. Minus the Login and Menu applications.

Per-Application Security Module Overview

Making use of the security modules, you can implement a complete access rules for systems developed by ScriptCase. The process for the security module creation is quick and simple.

Security module per Application

It has a restriction of access to applications according to the user. In this type of security, the administrator must define which applications can be accessed by system users.

In this type of security, these tables are created:

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
"app_name" TEXT NOT NULL,
"app_type" TEXT,
"description" TEXT,
PRIMARY KEY ("app_name")
)
```

sec_users_apps

```
CREATE TABLE "sec_users_apps" (
"login" TEXT NOT NULL,
"app_name" TEXT NOT NULL,
"priv_access" TEXT,
"priv_insert" TEXT,
"priv_delete" TEXT,
"priv_update" TEXT,
"priv_export" TEXT,
"priv_print" TEXT,
PRIMARY KEY ("login", "app_name")
)
```

sec_logged

```
CREATE TABLE "sec_logged" (
    "login" TEXT NOT NULL,
    "date_login" TEXT,
    "sc_session" TEXT,
    "ip" TEXT
);
```

> The table **Logged users** will be created only if the option **Protect Logged users** is checked during the Security Module creation.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

The table **users_social** will be created only if the option **Use social networks** is checked during the Security Module creation.

sec_settings

```
CREATE TABLE "sec_settings" (
set_name TEXT NOT NULL,
set_value TEXT,
PRIMARY KEY ("set_name")
)
```

Use Existing Tables

Connection

Define the connection where the security module tables fixed. All existing connections in the project will be displayed

> If you need to create a new connection, click here and see how.

Use existing tables

Allows you to use existing tables in your database. These tables must have, at least, the same fields used by the Scriptcase security module.

> We recommend that you use this option if you have used Scriptcase to create the tables before, to minimize errors.

If you want to know the specifications for this security module, click here.

Create tables

Once this option is selected, Scriptcase is responsible for creating the tables necessary for the use of the security module.

Tables prefix

Defines a prefix for the tables that will be created by the security module.

> By default, the Scriptcase uses "**sec**_".

Delete if tables already exist

This option deletes tables with the same name from your database.

E.G.: When defining the **prj**_ prefix for your tables, if there is any table with the name **prj_users** in your database, it will be deleted.

> This option is only available when selecting a **Create tables** option.

Protect Logged Users

This option prevents the same user from accessing the system simultaneously in different session.

By checking this option, the **logged** table will be created.

> When using the default prefix or the table name will be **sec_logged**.

Use Social Networks

This option allows the configuration of Facebook and Twitter for the authentication of system users.

By checking this option, the **users_social** table will be created.

> When using Social Networks, the **Login Template** option will not be available to define the security module login layout.

Association tables

This step is essential if the option **“Use existing tables”** is selected.

In this case, you must associate the fields from the existing tables to fields of security applications (applications generated by the Security Module).

> If you have selected the option **“Create tables”**, in the previous step, it ssociates the fields automatically.

If you choose to create the tables beforehand, see below as local tables and fields.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
"app_name" TEXT NOT NULL,
"app_type" TEXT,
"description" TEXT,
PRIMARY KEY ("app_name")
)
```

sec_users_apps

```
CREATE TABLE "sec_users_apps" (
"login" TEXT NOT NULL,
"app_name" TEXT NOT NULL,
"priv_access" TEXT,
"priv_insert" TEXT,
"priv_delete" TEXT,
"priv_update" TEXT,
"priv_export" TEXT,
"priv_print" TEXT,
PRIMARY KEY ("login", "app_name")
)
```

sec_logged

```
CREATE TABLE "sec_logged" (
    "login" TEXT NOT NULL,
    "date_login" TEXT,
    "sc_session" TEXT,
    "ip" TEXT
);
```

> The table **Logged users** will be created only if the option **Protect Logged users** is checked during the Security Module creation.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
```

```
);
```

The table **users_social** will be created only if the option **Use social networks** is checked during the Security Module creation.

General

Applications Prefix

You can set a prefix to the application names of the Security Module.

When session expires

Define the behaviour when the session expires.

- **No action** - The user continues using the application, but no saves after the session expires.
- **Redirect to login after the session expiration** - The application returns to the login after expiring the session.
- **Display a message that the session has expired** - Shows the message “session has expired” to the user.

Encryption

Use encryption to store the password in the table of users.

Enable Security

Activate the flag Application Security for all project applications.

Remember login

Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.

Cookie expiration time

Cookie expiration time in days for **Remember login** option

Label Position at Login

Defines the Label positioning of the fields in relation to the data.

- **Beside** - Default value of the applications, placing the label on the right side of the input label to the Side
- **Above** - Positions the label above the input label Above
- **Below** - Positions the label below the input label Below
- **Watermark** - Positions the label as Watermark.

Enable Captcha

It activates the captcha for the login application.

- **No** - Does not display the captcha in the login application.
- **Captcha** - Uses the built-in scriptcase library for captcha display.
- **reCAPTCHA** - Uses Google’s reCAPTCHA V2. To configure, click here.

  Watch a video of reCAPTCHA. If you prefer, see a tutorial on how to generate the keys.

Site Key

Key generated by Google after reCAPTCHA project creation.

Secret Key

Key generated by Google after reCAPTCHA project creation.

Folder

The project folder name to store the applications generated by Security Module.

Theme

The theme to create the Security Module applications.

Log

This option is available if the project already has a Log Module.

> Click here and check out how to create a Log Module

Menu

This option is only available if your project already has a Menu Application. You can associate the existing Menu and include all applications generated by the Security Module to it. If you do not select an existing Menu here, it creates a new Menu Application.

Menu Type

Security Module menu type (option only available if you do not select an existing menu in the previous item)

> This option is available when using the **Create Application Menu** option in the **menu** item

Use SweetAlert

Enables the use of sweetAlert in security module applications.

Settings

Defines whether the security module editing application will be generated.

With this option active, the system administrator user will be able to change some settings of the security module at the end, when accessing the system.

See below for the list of options available for editing the security module.

- **Action for session expiration** - Define the system behavior when the session expires. This feature is triggered as soon as a new action is performed.
  - **No action:** No action will be taken and the user will continue using the system, however no action will be saved.
  - **Redirect to login:** When performing some action on the system, the user will be redirected to the login application.
  - **Displays a message and redirects to login:** or perform some action on the system, a message will be displayed informing you that the session has expired with an ok button. By clicking on the button, the user will be redirected to the login application.
- **Remember login** - Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.
- **Remember login** expiration time - Cookie expiration time in days for the **Remember login** option. By default, Scriptcase sets the limit to 30 days.
- **Recover Password** - Enables the reset password button on the login screen.
- **New Users** - Enables the add new users button on the login screen.
- **Brute force** - Enables or disables the failed access attempt limit.
- **Block time for Brute Force protection** - Defines the time, in minutes, that the user will remain unreachable after several failed login attempts.
- **Brute Force Attempts for blocking** - Defines the number of failed access attempts until protection is activated.
- **2F** - Defines whether two-factor authentication will be used on the system.
- **2FA Expiration Time** - Determines the time in seconds that the 2FA authentication token will expire

Login

Here you can set the type and amount of characters allowed for the username and password fields.

User

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

Password

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

View password characters

This option toggles in the password field, allowing the displayed password to be displayed.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

E-mail Settings

Email Server

Select an email API to use in the security module.

This email will be used in the options:

- All options of **Retrieve Password**
- **New User** when select the options: **Requires activation by email**, **Send email to administrator**.

Custon - Enter the SMTP data in the options below.

> The options below are only available when the **custon** option is selected.

SMTP Server

Enter the SMTP server address.

SMTP Port

Enter the SMTP server port. This information must comply with the secure connection option. Use 465 for SSL, 587 for TLS, or 25 for unsafe connection. If you do not inform the port, Scriptcase applies the default one: 25.

Secure Connection

Use SSL or TSL, or leave it blank for insecure connection.

- SSL
- TSL

SMTP User

Enter the SMTP User information.

SMTP Password

Enter the SMTP password information.

SMTP E-mail

Enter the SMTP outgoing email.

SMTP

> To proceed with the password recovery, it is necessary that has the SMTP server configured for a successful process. Look here how to realize this procedure.

Yahoo

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.
> See also: Ways to securely access Yahoo Mail

SMTP Server Settings Yahoo	
SMTP Server	smtp.mail.yahoo.com
Port	465 or 587
E-mail address	Your full email address (name@domain.com.)

SMTP Server Settings Yahoo	
Password	The password for your account.
Requires SSL	Yes
Requires TSL	Yes (if available)
Requires authentication	Yes

Gmail

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.

SMTP Server Settings Gmail	
SMTP Server	smtp.gmail.com
E-mail address	Your full email address (Ex. example@gmail.com)
Password	The password for your account.
Port (TLS)	587
Port (SSL)	465
TLS / SSL Required	Yes

Outlook / Hotmail

For more information about these settings Click Here

SMTP Server Settings Outlook.com	
SMTP Server	smtp-mail.outlook.com
E-mail address	Your full email address (Ex. myname@outlook.com, not an alias)
Password	The password for your account.
Port	587 (you can use port 25 as an alternative)
TLS / SSL Required	Yes

Retrieve Password

It defines the password recovery method in the project.

> The password recovery options work only when the SMTP server is configured, please check the Email settings.

Send the password by e-mail

The system emails the user password just if the SMTP has been correctly set up, and the password is not using encryption.

Reset password and send new one by email

The system resets the password automatically and sends it to the user by email (just if the SMTP has been correctly set up).

Send a link to e-mail with reset option

The system emails the user a link to access an application and reset the password.

New users

Settings for creating new users in the security system.

> The options activation by email and send an email to admin work only when the SMTP server is configured properly, access Email settings to set it up.

Allows the registration of new users

This option sets the availability of users registration directly in the login system. If you do not check this option, only users with administrative access can insert new users.

Requires activation by email

This option sets whether the new user must validate his registration by email to activate the account (Configure the email SMTP to use this option)

Send email to administrator

This option sets whether the system administrator receives an email whenever a new user is registered. (Configure the email SMTP to use this option)

Social Network

Allows the use of **Facebook** and **Twitter** to login to the security module.

> When using social network authentication the **Login Template** option will not be available.

Facebook

Enable login via facebook. For the login to work, it is necessary to configure an application to obtain the **App ID** and **Secret**

See how to generate facebook credentials

App ID

Enter the **Application ID** available in the application settings on facebook developers.

Secret

Enter the **Application Secret Key** available in the application settings on facebook developers.

Twitter

Enable login via Twitter. For the login to work, it is necessary to configure an application to obtain the **App ID** and **Secret**

App ID

Enter the **Application ID**.

Secret

Enter the **Application Secret Key**

Logged Users

Defines the system behavior for login protection.

This option is available if you have checked Protect logged users during the connection step.

Display logged users

If you check this option Scriptcase will also create with the Security Module a Grid Application to display a report with all users current logged on the system

Brute Force Attack Protection

Enables/disables the blocking of users after some unsuccessful access attempts.

Brute Force lockout time (in Minutes)

Time, in minutes, that the user will remain inaccessible after several unsuccessful access attempts. (Available only when rute Force Attack Protection is enabled)

Numbers of attempts before lock

Number of failed access attempts, until the protection is enabled. (Available only when enable protection for brute force attacks)

Login Template

Allows the use of a custom HTML when creating the security module login application.

This HTML must be previously loaded into an external library, which in turn must be enabled for use in the project. See how to enable an external library.

> This option is only available when **Authentication by social network** is not being used.

Use the template for the lofin screen

Defines whether the login application will use a custom HTML or a standard Scriptcase theme.

No

By checking no, the login application will be created with the project's default theme.

Yes

When checking yes, the **Login Template** button will be enabled, thus allowing the selection of one of the listed templates.

> It is necessary to enable the external library that will be used.

- **Enable external Libraries** - Redirects to external library management screen.
- **Reload** - Reloads the select, listing the unlisted libraries after enabled.

Two Factor Authentication

Authentication (2FA)

In this option the user will select the type of 2 factor authentication to be used, there are three options:

Auth

This option will send the code to your Authentication app from Google.

> See how to use the authentication API

SMS

This option will send the authentication code by SMS.

> See how to use the SMS API

E-mail

This option will send the authentication code by SMS.

> See how to use the email sending APIs

API

In this option you select the API previously created to be used to send the code.

Code expiration time

Time in following to expire token sent for authentication.

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the security system and add new users.

> ATTENTION: To increase the security of your project, we recommend changing the default password “admin/admin” at this step or after the first access.

Login

This option sets the system administrator user.

> By default the defined user is **admin**

Password

This option sets the system administrator password.

> By default the defined password is **admin**

Name

This option sets the name of the administrator user on the system.

E-mail

This option sets the system administrator user E-mail.

> For sending email to the administrator to work, either SMTP is configured correctly or an email API is selected in the Email Settings screen in the previous step.

Add Applications

Add Applications

This option adds the applications already created in the project to the Security Module applications' table.

Application Includes - When selected, all existing applications in the project will be included in the security module.

When unchecking this option, the existing applications in the project when creating the module will need to be inserted through the application synchronization, after executing the security module.

Save Profile

Using this option, you save all settings during the Security Module creation. It can be used later for other projects.e

Save Profile

Allows you to save a profile with all the current security module settings.

Name

Profile name. It identifies the profile.

Target

This option sets what developers can use the saved profile afterward.

- Public - Set the security profile available in any project of your Scriptcase.
- Project - Set the security profile available only in the current project.
- User - Set the security profile available only to the current Scriptcase user

Building Security

Security module creation summary screen, listing changes made by the tool.

Open Project

Go to the Home of the currently open project.

Generate Source Code

Generates the source code of all applications created by the security module.

> After the creation of the security module, it will be necessary to generate the phone of all the applications, which all had a change in the use security flag. Minus the Login and Menu applications.

Per-Group Security Module Overview

Making use of the security modules, you can implement a complete access rules for systems developed by ScriptCase. The process for the security module creation is quick and simple.

Security module per Group

In this type of security, access to applications is defined according to the user groups created and configured by the system administrator.

> Every user must be part of at least one user group.

In this type of security, these tables are created:

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
"app_name" TEXT NOT NULL,
"app_type" TEXT,
"description" TEXT,
PRIMARY KEY ("app_name")
)
```

sec_groups

```
CREATE TABLE "sec_groups" (
    "group_id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    "description" TEXT,
    UNIQUE ("description")
)
```

sec_users_groups

```
CREATE TABLE "sec_users_groups" (
    "login" TEXT NOT NULL,
    "group_id" INTEGER NOT NULL,
    PRIMARY KEY ("login", "group_id")
)
```

sec_groups_apps

```
CREATE TABLE "sec_groups_apps" (
    "group_id" INTEGER NOT NULL,
    "app_name" TEXT NOT NULL,
    "priv_access" TEXT,
    "priv_insert" TEXT,
    "priv_delete" TEXT,
    "priv_update" TEXT,
    "priv_export" TEXT,
    "priv_print" TEXT,
    PRIMARY KEY ("group_id", "app_name")
)
```

sec_logged

```
CREATE TABLE "sec_logged" (
    "login" TEXT NOT NULL,
    "date_login" TEXT,
    "sc_session" TEXT,
    "ip" TEXT
);
```

The table **Logged users** will be created only if the option **Protect Logged users** is checked during the Security Module creation.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

The table **users_social** will be created only if the option **Use social networks** is checked during the Security Module creation.

sec_settings

```
CREATE TABLE "sec_settings" (
set_name TEXT NOT NULL,
set_value TEXT,
PRIMARY KEY ("set_name")
)
```

Use Existing Tables

Connection

Define the connection where the security module tables fixed. All existing connections in the project will be displayed

> If you need to create a new connection, click here and see how.

Use existing tables

Allows you to use existing tables in your database. These tables must have, at least, the same fields used by the Scriptcase security module.

> We recommend that you use this option if you have used Scriptcase to create the tables before, to minimize errors.

If you want to know the specifications for this security module, click here.

Create tables

Once this option is selected, Scriptcase is responsible for creating the tables necessary for the use of the security module.

Tables prefix

Defines a prefix for the tables that will be created by the security module.

> By default, the Scriptcase uses "**sec**_".

Delete if tables already exist

This option deletes tables with the same name from your database.

E.G.: When defining the **prj**_ prefix for your tables, if there is any table with the name **prj_users** in your database, it will be deleted.

> This option is only available when selecting a **Create tables** option.

Protect Logged Users

This option prevents the same user from accessing the system simultaneously in different session.

By checking this option, the **logged** table will be created.

> When using the default prefix or the table name will be **sec_logged**.

Use Social Networks

This option allows the configuration of Facebook and Twitter for the authentication of system users.

By checking this option, the **users_social** table will be created.

> When using Social Networks, the **Login Template** option will not be available to define the security module login layout.

Association tables

This step is essential if the option **“Use existing tables”** is selected.

In this case, you must associate the fields from the existing tables to fields of security applications (applications generated by the Security Module).

> If you have selected the option **“Create tables”**, in the previous step, it ssociates the fields automatically.

If you choose to create the tables beforehand, see below as local tables and fields.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
"app_name" TEXT NOT NULL,
"app_type" TEXT,
"description" TEXT,
PRIMARY KEY ("app_name")
)
```

sec_groups

```
CREATE TABLE "sec_groups" (
    "group_id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    "description" TEXT,
    UNIQUE ("description")
)
```

sec_users_groups

```
CREATE TABLE "sec_users_groups" (
    "login" TEXT NOT NULL,
    "group_id" INTEGER NOT NULL,
    PRIMARY KEY ("login", "group_id")
)
```

sec_groups_apps

```
CREATE TABLE "sec_groups_apps" (
    "group_id" INTEGER NOT NULL,
    "app_name" TEXT NOT NULL,
    "priv_access" TEXT,
    "priv_insert" TEXT,
    "priv_delete" TEXT,
    "priv_update" TEXT,
    "priv_export" TEXT,
    "priv_print" TEXT,
    PRIMARY KEY ("group_id", "app_name")
)
```

sec_logged

```
CREATE TABLE "sec_logged" (
```

```
    "login" TEXT NOT NULL,
    "date_login" TEXT,
    "sc_session" TEXT,
    "ip" TEXT
);
```

The table **Logged users** will be created only if the option **Protect Logged users** is checked during the Security Module creation.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

The table **users_social** will be created only if the option **Use social networks** is checked during the Security Module creation.

General

Applications Prefix

You can set a prefix to the application names of the Security Module.

When session expires

Define the behaviour when the session expires.

- **No action** - The user continues using the application, but no saves after the session expires.
- **Redirect to login after the session expiration** - The application returns to the login after expiring the session.
- **Display a message that the session has expired** - Shows the message “session has expired” to the user.

Encryption

Use encryption to store the password in the table of users.

Enable Security

Activate the flag Application Security for all project applications.

Remember login

Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.

Cookie expiration time

Cookie expiration time in days for **Remember login** option

Label Position at Login

Defines the Label positioning of the fields in relation to the data.

- **Beside** - Default value of the applications, placing the label on the right side of the input label to the Side
- **Above** - Positions the label above the input label Above
- **Below** - Positions the label below the input label Below
- **Watermark** - Positions the label as Watermark.

Enable Captcha

It activates the captcha for the login application.

- **No** - Does not display the captcha in the login application.
- **Captcha** - Uses the built-in scriptcase library for captcha display.
- **reCAPTCHA** - Uses Google’s reCAPTCHA V2. To configure, click here.

  Watch a video of reCAPTCHA. If you prefer, see a tutorial on how to generate the keys.

Site Key

Key generated by Google after reCAPTCHA project creation.

Secret Key

Key generated by Google after reCAPTCHA project creation.

Folder

The project folder name to store the applications generated by Security Module.

Theme

The theme to create the Security Module applications.

Log

This option is available if the project already has a Log Module.

> Click here and check out how to create a Log Module

Menu

This option is only available if your project already has a Menu Application. You can associate the existing Menu and include all applications generated by the Security Module to it. If you do not select an existing Menu here, it creates a new Menu Application.

Menu Type

Security Module menu type (option only available if you do not select an existing menu in the previous item)

> This option is available when using the **Create Application Menu** option in the **menu** item

Use SweetAlert

Enables the use of sweetAlert in security module applications.

Settings

Defines whether the security module editing application will be generated.

With this option active, the system administrator user will be able to change some settings of the security module at the end, when accessing the system.

See below for the list of options available for editing the security module.

- **Action for session expiration** - Define the system behavior when the session expires. This feature is triggered as soon as a new action is performed.
  - **No action:** No action will be taken and the user will continue using the system, however no action will be saved.
  - **Redirect to login:** When performing some action on the system, the user will be redirected to the login application.
  - **Displays a message and redirects to login:** or perform some action on the system, a message will be displayed informing you that the session has expired with an ok button. By clicking on the button, the user will be redirected to the login application.
- **Remember login** - Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.
- **Remember login** expiration time - Cookie expiration time in days for the **Remember login** option. By default, Scriptcase sets the limit to 30 days.
- **Recover Password** - Enables the reset password button on the login screen.
- **New Users** - Enables the add new users button on the login screen.
- **Brute force** - Enables or disables the failed access attempt limit.
- **Block time for Brute Force protection** - Defines the time, in minutes, that the user will remain unreachable after several failed login attempts.
- **Brute Force Attempts for blocking** - Defines the number of failed access attempts until protection is activated.
- **2F** - Defines whether two-factor authentication will be used on the system.
- **2FA Expiration Time** - Determines the time in seconds that the 2FA authentication token will expire

Login

Here you can set the type and amount of characters allowed for the username and password fields.

User

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

Password

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

View password characters

This option toggles in the password field, allowing the displayed password to be displayed.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

E-mail Settings

Email Server

Select an email API to use in the security module.

This email will be used in the options:

- All options of **Retrieve Password**
- **New User** when select the options: **Requires activation by email**, **Send email to administrator**.

Custon - Enter the SMTP data in the options below.

> The options below are only available when the **custon** option is selected.

SMTP Server

Enter the SMTP server address.

SMTP Port

Enter the SMTP server port. This information must comply with the secure connection option. Use 465 for SSL, 587 for TLS, or 25 for unsafe connection. If you do not inform the port, Scriptcase applies the default one: 25.

Secure Connection

Use SSL or TSL, or leave it blank for insecure connection.

- SSL
- TSL

SMTP User

Enter the SMTP User information.

SMTP Password

Enter the SMTP password information.

SMTP E-mail

Enter the SMTP outgoing email.

SMTP

> To proceed with the password recovery, it is necessary that has the SMTP server configured for a successful process. Look here how to realize this procedure.

Yahoo

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.
> See also: Ways to securely access Yahoo Mail

SMTP Server Settings Yahoo	
SMTP Server	smtp.mail.yahoo.com
Port	465 or 587
E-mail address	Your full email address (name@domain.com.)

SMTP Server Settings Yahoo	
Password	The password for your account.
Requires SSL	Yes
Requires TSL	Yes (if available)
Requires authentication	Yes

Gmail

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.

SMTP Server Settings Gmail	
SMTP Server	smtp.gmail.com
E-mail address	Your full email address (Ex. example@gmail.com)
Password	The password for your account.
Port (TLS)	587
Port (SSL)	465
TLS / SSL Required	Yes

Outlook / Hotmail

For more information about these settings Click Here

SMTP Server Settings Outlook.com	
SMTP Server	smtp-mail.outlook.com
E-mail address	Your full email address (Ex. myname@outlook.com, not an alias)
Password	The password for your account.
Port	587 (you can use port 25 as an alternative)
TLS / SSL Required	Yes

Retrieve Password

It defines the password recovery method in the project.

> The password recovery options work only when the SMTP server is configured, please check the Email settings.

Send the password by e-mail

The system emails the user password just if the SMTP has been correctly set up, and the password is not using encryption.

Reset password and send new one by email

The system resets the password automatically and sends it to the user by email (just if the SMTP has been correctly set up).

Send a link to e-mail with reset option

The system emails the user a link to access an application and reset the password.

New users

Settings for creating new users in the security system.

> The options activation by email and send an email to admin work only when the SMTP server is configured properly, access Email settings to set it up.

Allows the registration of new users

This option sets the availability of users registration directly in the login system. If you do not check this option, only users with administrative access can insert new users.

Requires activation by email

This option sets whether the new user must validate his registration by email to activate the account (Configure the email SMTP to use this option)

Send email to administrator

This option sets whether the system administrator receives an email whenever a new user is registered. (Configure the email SMTP to use this option)

Social Network

Allows the use of **Facebook** and **Twitter** to login to the security module.

> When using social network authentication the **Login Template** option will not be available.

Facebook

Enable login via facebook. For the login to work, it is necessary to configure an application to obtain the **App ID** and **Secret**

See how to generate facebook credentials

App ID

Enter the **Application ID** available in the application settings on facebook developers.

Secret

Enter the **Application Secret Key** available in the application settings on facebook developers.

Twitter

Enable login via Twitter. For the login to work, it is necessary to configure an application to obtain the **App ID** and **Secret**

App ID

Enter the **Application ID**.

Secret

Enter the **Application Secret Key**

Logged Users

Defines the system behavior for login protection.

This option is available if you have checked Protect logged users during the connection step.

Display logged users

If you check this option Scriptcase will also create with the Security Module a Grid Application to display a report with all users current logged on the system

Brute Force Attack Protection

Enables/disables the blocking of users after some unsuccessful access attempts.

Brute Force lockout time (in Minutes)

Time, in minutes, that the user will remain inaccessible after several unsuccessful access attempts. (Available only when rute Force Attack Protection is enabled)

Numbers of attempts before lock

Number of failed access attempts, until the protection is enabled. (Available only when enable protection for brute force attacks)

Login Template

Allows the use of a custom HTML when creating the security module login application.

This HTML must be previously loaded into an external library, which in turn must be enabled for use in the project. See how to enable an external library.

> This option is only available when **Authentication by social network** is not being used.

Use the template for the lofin screen

Defines whether the login application will use a custom HTML or a standard Scriptcase theme.

No

By checking no, the login application will be created with the project's default theme.

Yes

When checking yes, the **Login Template** button will be enabled, thus allowing the selection of one of the listed templates.

> It is necessary to enable the external library that will be used.

- **Enable external Libraries** - Redirects to external library management screen.
- **Reload** - Reloads the select, listing the unlisted libraries after enabled.

Two Factor Authentication

Authentication (2FA)

In this option the user will select the type of 2 factor authentication to be used, there are three options:

Auth

This option will send the code to your Authentication app from Google.

> See how to use the authentication API

SMS

This option will send the authentication code by SMS.

> See how to use the SMS API

E-mail

This option will send the authentication code by SMS.

> See how to use the email sending APIs

API

In this option you select the API previously created to be used to send the code.

Code expiration time

Time in following to expire token sent for authentication.

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the security system and add new users.

> ATTENTION: To increase the security of your project, we recommend changing the default password “admin/admin” at this step or after the first access.

Login

This option sets the system administrator user.

> By default the defined user is **admin**

Password

This option sets the system administrator password.

> By default the defined password is **admin**

Name

This option sets the name of the administrator user on the system.

E-mail

This option sets the system administrator user E-mail.

> For sending email to the administrator to work, either SMTP is configured correctly or an email API is selected in the Email Settings screen in the previous step.

Add Applications

Add Applications

This option adds the applications already created in the project to the Security Module applications' table.

Application Includes - When selected, all existing applications in the project will be included in the security module.

When unchecking this option, the existing applications in the project when creating the module will need to be inserted through the application synchronization, after executing the security module.

Save Profile

Using this option, you save all settings during the Security Module creation. It can be used later for other projects.e

Save Profile

Allows you to save a profile with all the current security module settings.

Name

Profile name. It identifies the profile.

Target

This option sets what developers can use the saved profile afterward.

- Public - Set the security profile available in any project of your Scriptcase.
- Project - Set the security profile available only in the current project.
- User - Set the security profile available only to the current Scriptcase user

Building Security

Security module creation summary screen, listing changes made by the tool.

Open Project

Go to the Home of the currently open project.

Generate Source Code

Generates the source code of all applications created by the security module.

> After the creation of the security module, it will be necessary to generate the phone of all the applications, which all had a change in the use security flag. Minus the Login and Menu applications.

LDAP Auth-Only Security Module Overview

Making use of the security modules, you can implement a complete access rules for systems developed by ScriptCase. The process for the security module creation is quick and simple.

Security module per LDAP - Only Authenticate

In user Ldap Only Authenticate, all registered users have access to system applications, working only as user authentication.

This type of security does not need the creation of any table, having all user access control being performed by the Ldap server.

Use Existing Tables

Connection

Defines the connection that will be used in security module applications.

> This type of security does not generate any tables, as all control is performed through Ldap

General

Applications Prefix

You can set a prefix to the application names of the Security Module.

When session expires

Define the behaviour when the session expires.

- **No action** - The user continues using the application, but no saves after the session expires.
- **Redirect to login after the session expiration** - The application returns to the login after expiring the session.
- **Display a message that the session has expired** - Shows the message “session has expired” to the user.

Encryption

Use encryption to store the password in the table of users.

Enable Security

Activate the flag Application Security for all project applications.

Remember login

Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.

Cookie expiration time

Cookie expiration time in days for **Remember login** option

Label Position at Login

Defines the Label positioning of the fields in relation to the data.

- **Beside** - Default value of the applications, placing the label on the right side of the input label to the Side
- **Above** - Positions the label above the input label Above
- **Below** - Positions the label below the input label Below
- **Watermark** - Positions the label as Watermark.

Enable Captcha

It activates the captcha for the login application.

- **No** - Does not display the captcha in the login application.
- **Captcha** - Uses the built-in scriptcase library for captcha display.
- **reCAPTCHA** - Uses Google’s reCAPTCHA V2. To configure, click here.

  Watch a video of reCAPTCHA. If you prefer, see a tutorial on how to generate the keys.

Site Key

Key generated by Google after reCAPTCHA project creation.

Secret Key

Key generated by Google after reCAPTCHA project creation.

Folder

The project folder name to store the applications generated by Security Module.

Theme

The theme to create the Security Module applications.

Log

This option is available if the project already has a Log Module.

> Click here and check out how to create a Log Module

Menu

This option is only available if your project already has a Menu Application. You can associate the existing Menu and include all applications generated by the Security Module to it. If you do not select an existing Menu here, it creates a new Menu Application.

Menu Type

Security Module menu type (option only available if you do not select an existing menu in the previous item)

> This option is available when using the **Create Application Menu** option in the **menu** item

Use SweetAlert

Enables the use of sweetAlert in security module applications.

Settings

Defines whether the security module editing application will be generated.

With this option active, the system administrator user will be able to change some settings of the security module at the end, when accessing the system.

See below for the list of options available for editing the security module.

- **Action for session expiration** - Define the system behavior when the session expires. This feature is triggered as soon as a new action is performed.
  - **No action:** No action will be taken and the user will continue using the system, however no action will be saved.
  - **Redirect to login:** When performing some action on the system, the user will be redirected to the login application.
  - **Displays a message and redirects to login:** or perform some action on the system, a message will be displayed informing you that the session has expired with an ok button. By clicking on the button, the user will be redirected to the login application.
- **Remember login** - Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.
- **Remember login** expiration time - Cookie expiration time in days for the **Remember login** option. By default, Scriptcase sets the limit to 30 days.
- **Recover Password** - Enables the reset password button on the login screen.
- **New Users** - Enables the add new users button on the login screen.
- **Brute force** - Enables or disables the failed access attempt limit.
- **Block time for Brute Force protection** - Defines the time, in minutes, that the user will remain unreachable after several failed login attempts.
- **Brute Force Attempts for blocking** - Defines the number of failed access attempts until protection is activated.
- **2F** - Defines whether two-factor authentication will be used on the system.
- **2FA Expiration Time** - Determines the time in seconds that the 2FA authentication token will expire

Login

Here you can set the type and amount of characters allowed for the username and password fields.

User

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

Password

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

View password characters

This option toggles in the password field, allowing the displayed password to be displayed.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

Ldap

In Ldap settings it is necessary to inform the access data to the previously configured server.

Server

Enter here the LDAP server IP.

DN

Enter the input attributes.

Port

Enter here the port for access to the server. Default port is 389.

Sufix

Enter the user suffix.

Login Template

Allows the use of a custom HTML when creating the security module login application.

This HTML must be previously loaded into an external library, which in turn must be enabled for use in the project. See how to enable an external library.

> This option is only available when **Authentication by social network** is not being used.

Use the template for the lofin screen

Defines whether the login application will use a custom HTML or a standard Scriptcase theme.

No

By checking no, the login application will be created with the project's default theme.

Yes

When checking yes, the **Login Template** button will be enabled, thus allowing the selection of one of the listed templates.

> It is necessary to enable the external library that will be used.

- **Enable external Libraries** - Redirects to external library management screen.
- **Reload** - Reloads the select, listing the unlisted libraries after enabled.

Two Factor Authentication

Authentication (2FA)

In this option the user will select the type of 2 factor authentication to be used, there are three options:

Auth

This option will send the code to your Authentication app from Google.

> See how to use the authentication API

SMS

This option will send the authentication code by SMS.

> See how to use the SMS API

E-mail

This option will send the authentication code by SMS.

> See how to use the email sending APIs

API

In this option you select the API previously created to be used to send the code.

Code expiration time

Time in following to expire token sent for authentication.

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the security system and add new users.

> ATTENTION: To increase the security of your project, we recommend changing the default password “admin/admin” at this step or after the first access.

Login

This option sets the system administrator user.

> By default the defined user is **admin**

Password

This option sets the system administrator password.

> By default the defined password is **admin**

Name

This option sets the name of the administrator user on the system.

E-mail

This option sets the system administrator user E-mail.

> For sending email to the administrator to work, either SMTP is configured correctly or an email API is selected in the Email Settings screen in the previous step.

Save Profile

Using this option, you save all settings during the Security Module creation. It can be used later for other projects.e

Save Profile

Allows you to save a profile with all the current security module settings.

Name

Profile name. It identifies the profile.

Target

This option sets what developers can use the saved profile afterward.

- Public - Set the security profile available in any project of your Scriptcase.
- Project - Set the security profile available only in the current project.
- User - Set the security profile available only to the current Scriptcase user

Building Security

Security module creation summary screen, listing changes made by the tool.

Open Project

Go to the Home of the currently open project.

Generate Source Code

Generates the source code of all applications created by the security module.

> After the creation of the security module, it will be necessary to generate the phone of all the applications, which all had a change in the use security flag. Minus the Login and Menu applications.

LDAP Total-Control Security Module Overview

Making use of the security modules, you can implement a complete access rules for systems developed by ScriptCase. The process for the security module creation is quick and simple.

Ldap - Total Control

In this type of security, access to applications is defined according to the user groups created and configured by the system administrator.

Below see all the tables created for this type of security.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
"app_name" TEXT NOT NULL,
"app_type" TEXT,
"description" TEXT,
PRIMARY KEY ("app_name")
)
```

sec_users_apps

```
CREATE TABLE "sec_users_apps" (
"login" TEXT NOT NULL,
"app_name" TEXT NOT NULL,
"priv_access" TEXT,
"priv_insert" TEXT,
"priv_delete" TEXT,
"priv_update" TEXT,
"priv_export" TEXT,
"priv_print" TEXT,
PRIMARY KEY ("login", "app_name")
)
```

sec_logged

```
CREATE TABLE "sec_logged" (
    "login" TEXT NOT NULL,
    "date_login" TEXT,
    "sc_session" TEXT,
    "ip" TEXT
);
```

> The table **Logged users** will be created only if the option **Protect Logged users** is checked during the Security Module creation.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

The table **users_social** will be created only if the option **Use social networks** is checked during the Security Module creation.

sec_settings

```
CREATE TABLE "sec_settings" (
set_name TEXT NOT NULL,
set_value TEXT,
PRIMARY KEY ("set_name")
)
```

Use Existing Tables

Connection

Define the connection where the security module tables fixed. All existing connections in the project will be displayed

> If you need to create a new connection, click here and see how.

Use existing tables

Allows you to use existing tables in your database. These tables must have, at least, the same fields used by the Scriptcase security module.

> We recommend that you use this option if you have used Scriptcase to create the tables before, to minimize errors.

If you want to know the specifications for this security module, click here.

Create tables

Once this option is selected, Scriptcase is responsible for creating the tables necessary for the use of the security module.

Tables prefix

Defines a prefix for the tables that will be created by the security module.

> By default, the Scriptcase uses "**sec**_".

Delete if tables already exist

This option deletes tables with the same name from your database.

E.G.: When defining the **prj**_ prefix for your tables, if there is any table with the name **prj_users** in your database, it will be deleted.

> This option is only available when selecting a **Create tables** option.

Protect Logged Users

This option prevents the same user from accessing the system simultaneously in different session.

By checking this option, the **logged** table will be created.

> When using the default prefix or the table name will be **sec_logged**.

Use Social Networks

This option allows the configuration of Facebook and Twitter for the authentication of system users.

By checking this option, the **users_social** table will be created.

> When using Social Networks, the **Login Template** option will not be available to define the security module login layout.

Association tables

This step is essential if the option **“Use existing tables”** is selected.

In this case, you must associate the fields from the existing tables to fields of security applications (applications generated by the Security Module).

> If you have selected the option **“Create tables”**, in the previous step, it ssociates the fields automatically.

If you choose to create the tables beforehand, see below as local tables and fields.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    PRIMARY KEY ("login")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    "login" TEXT NOT NULL,
    "date_login" TEXT,
    "sc_session" TEXT,
    "ip" TEXT
);
```

> The table **Logged users** will be created only if the option **Protect Logged users** is checked during the Security Module creation.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> The table **users_social** will be created only if the option **Use social networks** is checked during the Security Module creation.

General

Applications Prefix

You can set a prefix to the application names of the Security Module.

When session expires

Define the behaviour when the session expires.

- **No action** - The user continues using the application, but no saves after the session expires.
- **Redirect to login after the session expiration** - The application returns to the login after expiring the session.
- **Display a message that the session has expired** - Shows the message “session has expired” to the user.

Encryption

Use encryption to store the password in the table of users.

Enable Security

Activate the flag Application Security for all project applications.

Remember login

Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.

Cookie expiration time

Cookie expiration time in days for **Remember login** option

Label Position at Login

Defines the Label positioning of the fields in relation to the data.

- **Beside** - Default value of the applications, placing the label on the right side of the input label to the Side
- **Above** - Positions the label above the input label Above
- **Below** - Positions the label below the input label Below
- **Watermark** - Positions the label as Watermark.

Enable Captcha

It activates the captcha for the login application.

- **No** - Does not display the captcha in the login application.
- **Captcha** - Uses the built-in scriptcase library for captcha display.
- **reCAPTCHA** - Uses Google’s reCAPTCHA V2. To configure, click here.

  Watch a video of reCAPTCHA. If you prefer, see a tutorial on how to generate the keys.

Site Key

Key generated by Google after reCAPTCHA project creation.

Secret Key

Key generated by Google after reCAPTCHA project creation.

Folder

The project folder name to store the applications generated by Security Module.

Theme

The theme to create the Security Module applications.

Log

This option is available if the project already has a Log Module.

> Click here and check out how to create a Log Module

Menu

This option is only available if your project already has a Menu Application. You can associate the existing Menu and include all applications generated by the Security Module to it. If you do not select an existing Menu here, it creates a new Menu Application.

Menu Type

Security Module menu type (option only available if you do not select an existing menu in the previous item)

> This option is available when using the **Create Application Menu** option in the **menu** item

Use SweetAlert

Enables the use of sweetAlert in security module applications.

Settings

Defines whether the security module editing application will be generated.

With this option active, the system administrator user will be able to change some settings of the security module at the end, when accessing the system.

See below for the list of options available for editing the security module.

- **Action for session expiration** - Define the system behavior when the session expires. This feature is triggered as soon as a new action is performed.
  - **No action:** No action will be taken and the user will continue using the system, however no action will be saved.
  - **Redirect to login:** When performing some action on the system, the user will be redirected to the login application.
  - **Displays a message and redirects to login:** or perform some action on the system, a message will be displayed informing you that the session has expired with an ok button. By clicking on the button, the user will be redirected to the login application.
- **Remember login** - Activating the flag allows the user to remain logged in when returning to the system when the session ends, without having logged out.
- **Remember login** expiration time - Cookie expiration time in days for the **Remember login** option. By default, Scriptcase sets the limit to 30 days.
- **Recover Password** - Enables the reset password button on the login screen.
- **New Users** - Enables the add new users button on the login screen.
- **Brute force** - Enables or disables the failed access attempt limit.
- **Block time for Brute Force protection** - Defines the time, in minutes, that the user will remain unreachable after several failed login attempts.
- **Brute Force Attempts for blocking** - Defines the number of failed access attempts until protection is activated.
- **2F** - Defines whether two-factor authentication will be used on the system.
- **2FA Expiration Time** - Determines the time in seconds that the 2FA authentication token will expire

Login

Here you can set the type and amount of characters allowed for the username and password fields.

User

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

Password

Minimum size

Minimum amount of characters used by the user.

Maximum size

Maximum size of the characters used by the user.

Allowed characters

Define which characters are allowed when creating a password.

View password characters

This option toggles in the password field, allowing the displayed password to be displayed.

- **All** - Accepts all values
- **Select** - Allows you to select the allowed characters. The options are:
  - Letters
  - Numbers
  - Accent
  - Cedilla
  - Space
  - Dot
  - Comma

E-mail Settings

Email Server

Select an email API to use in the security module.

This email will be used in the options:

- All options of **Retrieve Password**
- **New User** when select the options: **Requires activation by email**, **Send email to administrator**.

Custon - Enter the SMTP data in the options below.

> The options below are only available when the **custon** option is selected.

SMTP Server

Enter the SMTP server address.

SMTP Port

Enter the SMTP server port. This information must comply with the secure connection option. Use 465 for SSL, 587 for TLS, or 25 for unsafe connection. If you do not inform the port, Scriptcase applies the default one: 25.

Secure Connection

Use SSL or TSL, or leave it blank for insecure connection.

- SSL
- TSL

SMTP User

Enter the SMTP User information.

SMTP Password

Enter the SMTP password information.

SMTP E-mail

Enter the SMTP outgoing email.

SMTP

> To proceed with the password recovery, it is necessary that has the SMTP server configured for a successful process. Look here how to realize this procedure.

Yahoo

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.
> See also: Ways to securely access Yahoo Mail

SMTP Server Settings Yahoo	
SMTP Server	smtp.mail.yahoo.com
Port	465 or 587
E-mail address	Your full email address (name@domain.com.)

SMTP Server Settings Yahoo	
Password	The password for your account.
Requires SSL	Yes
Requires TSL	Yes (if available)
Requires authentication	Yes

Gmail

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.

SMTP Server Settings Gmail	
SMTP Server	smtp.gmail.com
E-mail address	Your full email address (Ex. example@gmail.com)
Password	The password for your account.
Port (TLS)	587
Port (SSL)	465
TLS / SSL Required	Yes

Outlook / Hotmail

For more information about these settings Click Here

SMTP Server Settings Outlook.com	
SMTP Server	smtp-mail.outlook.com
E-mail address	Your full email address (Ex. myname@outlook.com, not an alias)
Password	The password for your account.
Port	587 (you can use port 25 as an alternative)
TLS / SSL Required	Yes

Ldap

In Ldap settings it is necessary to inform the access data to the previously configured server.

Server

Enter here the LDAP server IP.

DN

Enter the input attributes.

Port

Enter here the port for access to the server. Default port is 389.

Sufix

Enter the user suffix.

Social Network

Allows the use of **Facebook** and **Twitter** to login to the security module.

> When using social network authentication the **Login Template** option will not be available.

Facebook

Enable login via facebook. For the login to work, it is necessary to configure an application to obtain the **App ID** and **Secret**

See how to generate facebook credentials

App ID

Enter the **Application ID** available in the application settings on facebook developers.

Secret

Enter the **Application Secret Key** available in the application settings on facebook developers.

Twitter

Enable login via Twitter. For the login to work, it is necessary to configure an application to obtain the **App ID** and **Secret**

App ID

Enter the **Application ID**.

Secret

Enter the **Application Secret Key**

Logged Users

Defines the system behavior for login protection.

This option is available if you have checked Protect logged users during the connection step.

Display logged users

If you check this option Scriptcase will also create with the Security Module a Grid Application to display a report with all users current logged on the system

Brute Force Attack Protection

Enables/disables the blocking of users after some unsuccessful access attempts.

Brute Force lockout time (in Minutes)

Time, in minutes, that the user will remain inaccessible after several unsuccessful access attempts. (Available only when rute Force Attack Protection is enabled)

Numbers of attempts before lock

Number of failed access attempts, until the protection is enabled. (Available only when enable protection for brute force attacks)

Login Template

Allows the use of a custom HTML when creating the security module login application.

This HTML must be previously loaded into an external library, which in turn must be enabled for use in the project. See how to enable an external library.

> This option is only available when **Authentication by social network** is not being used.

Use the template for the lofin screen

Defines whether the login application will use a custom HTML or a standard Scriptcase theme.

No

By checking no, the login application will be created with the project's default theme.

Yes

When checking yes, the **Login Template** button will be enabled, thus allowing the selection of one of the listed templates.

> It is necessary to enable the external library that will be used.

- **Enable external Libraries** - Redirects to external library management screen.
- **Reload** - Reloads the select, listing the unlisted libraries after enabled.

Two Factor Authentication

Authentication (2FA)

In this option the user will select the type of 2 factor authentication to be used, there are three options:

Auth

This option will send the code to your Authentication app from Google.

> See how to use the authentication API

SMS

This option will send the authentication code by SMS.

> See how to use the SMS API

E-mail

This option will send the authentication code by SMS.

> See how to use the email sending APIs

API

In this option you select the API previously created to be used to send the code.

Code expiration time

Time in following to expire token sent for authentication.

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the security system and add new users.

> ATTENTION: To increase the security of your project, we recommend changing the default password “admin/admin” at this step or after the first access.

Login

This option sets the system administrator user.

> By default the defined user is **admin**

Password

This option sets the system administrator password.

> By default the defined password is **admin**

Name

This option sets the name of the administrator user on the system.

E-mail

This option sets the system administrator user E-mail.

> For sending email to the administrator to work, either SMTP is configured correctly or an email API is selected in the Email Settings screen in the previous step.

Add Applications

Add Applications

This option adds the applications already created in the project to the Security Module applications' table.

Application Includes - When selected, all existing applications in the project will be included in the security module.

When unchecking this option, the existing applications in the project when creating the module will need to be inserted through the application synchronization, after executing the security module.

Save Profile

Using this option, you save all settings during the Security Module creation. It can be used later for other projects.e

Save Profile

Allows you to save a profile with all the current security module settings.

Name

Profile name. It identifies the profile.

Target

This option sets what developers can use the saved profile afterward.

- Public - Set the security profile available in any project of your Scriptcase.
- Project - Set the security profile available only in the current project.
- User - Set the security profile available only to the current Scriptcase user

Building Security

Security module creation summary screen, listing changes made by the tool.

Open Project

Go to the Home of the currently open project.

Generate Source Code

Generates the source code of all applications created by the security module.

> After the creation of the security module, it will be necessary to generate the phone of all the applications, which all had a change in the use security flag. Minus the Login and Menu applications.

Security Module Overview

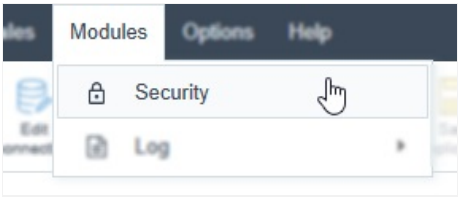
The **Security Module** automatically generates a set of applications used to manage system security. This generator creates all validation of access and use of applications, in addition to options such as: *password recovery*, *two-factor authentication*, *new user registration*, *login with social network*, among other resources available for settings.

Resources must be configured by the developer in the module creation process according to the system definition.

With the **New Security Module**, all settings defined at creation will also be available in the **Configuration** application, as long as it is enabled by the developer, thus allowing the system administrator user to configure/change the settings used by the developer when creating the module.

With the new module, changes such as email configuration, authentication API, password length, use of captcha, among other options, are now available within the system itself.

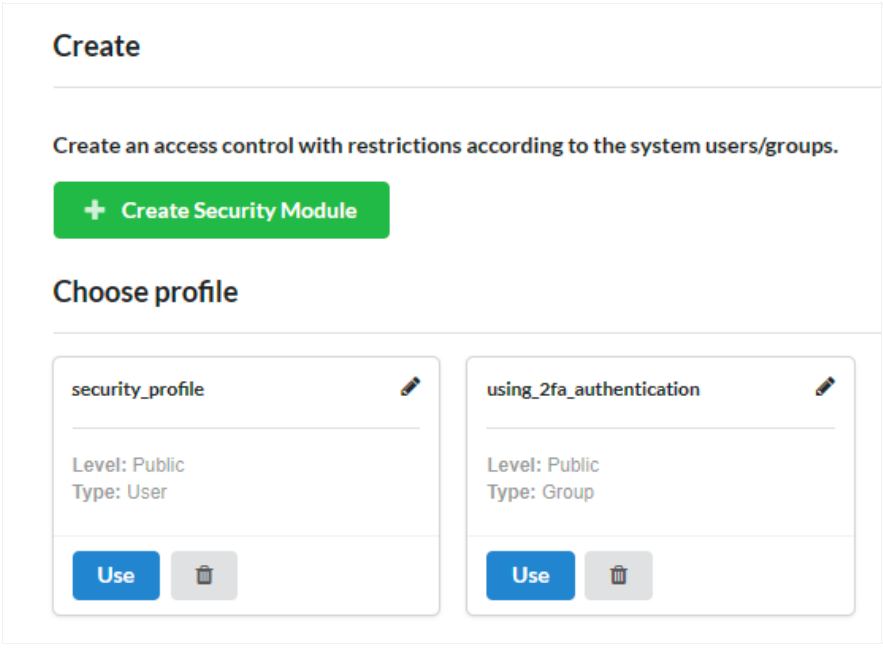
The security generator is available in the menu: **Modules > Security**.



Create/Choose Profile

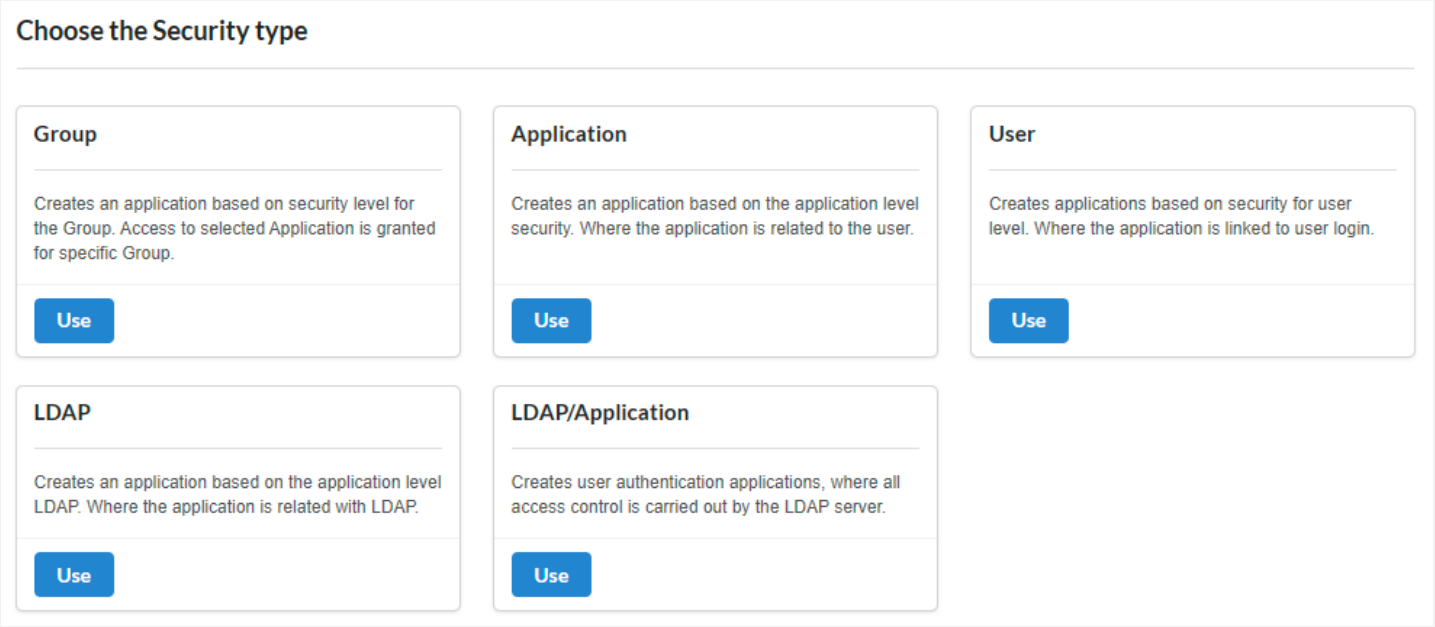
On this screen, the **Create Security Module** button (to create a new blank module) and the saved profiles of previously created modules will be displayed. These profiles store the configurations made when creating the module, so when selecting a profile the configuration options will be filled in.

The **Create/Choose Profile** page will be opened if there is a security profile saved, otherwise, the security module will be opened on the **Security Types** screen



Types of security

At this stage, the developer must select the type of security they want to create, according to the system's needs.



The types of security are: Group, Application, User, LDAP e LDAP/Application. They all provide similar configuration resources, the main difference between them is in the way access control and permission to use applications are applied to users.

Group

In this type of security, system access and use permissions will be assigned to groups of users, for example: *Administrators*, *Employees* and *Customers*.

Users will have access permissions according to the groups they are part of. A user in the Administrator group will have full access to the system (sales report, user control), while a user in the customer group would only have access to the store and profile configuration.

Click Here to access detailed documentation for the Group Security Module.

Application

Security by application works in a similar way to group security, with access permissions and use of the system, the difference is in the way these permissions will be applied. In this type of security, the administrator will have to define the permissions to use the system individually, per user.

Click Here to access the detailed documentation of the **Application Security Module**.

User

The module per user only has basic access control to the system. In this type of security, it is not possible to define access levels, all users will have access to all system applications.

Click Here to access detailed documentation of the **Security Module per User**.

LDAP

This security module works in a similar way to user security, only controlling access to the system, without access levels, but using an LDAP server.

Click Here to access detailed documentation for the **Ldap** Type Security Module.

LDAP/Application

This type of security with LDAP works like the group security module, where the administrator defines the access and use

permissions of the system for a group of users, but using an LDAP server.

Click Here to access detailed documentation for the Security Module of type **Ldap/Application**.

Group-Based Security Module

In the **Group Security Module**, the system administrator has the ability to restrict access to different applications based on predefined user groups. This means that each group can have specific permissions to access certain functionalities or areas of the system, offering precise control over the actions allowed for each set of users.

Check below the tables that will be created in your database when configuring a security module.

> The **sec_logged** (Logged User) and **sec_users_social** (Social Network Authentication) tables will be created if these features are enabled, the creation of the other tables is mandatory.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    "role" TEXT,
    "phone" TEXT,
    "pswd_last_updated" TIMESTAMP,
    "mfa_last_updated" TIMESTAMP DEFAULT NULL,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
    "app_name" TEXT NOT NULL,
    "app_type" TEXT,
    "description" TEXT,
    PRIMARY KEY ("app_name")
);
```

sec_groups

```
CREATE TABLE "sec_groups" (
    "group_id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    "description" TEXT
);
```

sec_users_groups

```
CREATE TABLE "sec_users_groups" (
    "login" TEXT NOT NULL,
    "group_id" INTEGER NOT NULL,
    PRIMARY KEY ("login", "group_id")
);
```

sec_groups_apps

```
CREATE TABLE "sec_groups_apps" (
    "group_id" INTEGER NOT NULL,
    "app_name" TEXT NOT NULL,
    "priv_access" TEXT,
    "priv_insert" TEXT,
    "priv_delete" TEXT,
    "priv_update" TEXT,
    "priv_export" TEXT,
    "priv_print" TEXT,
    PRIMARY KEY ("group_id", "app_name")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    login TEXT NOT NULL,
    date_login TEXT,
    sc_session TEXT,
    ip TEXT
);
```

> **NOTA:** A tabela **Usuários logados** será criada somente se a opção **Proteger usuários logados** for habilitada durante a criação do Módulo de Segurança.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> **NOTA:** A tabela **users_social** será criada somente se a opção **Utilizar redes sociais** for verificada durante a criação do Módulo de Segurança.

sec_settings

```
CREATE TABLE "sec_settings" (
    set_name TEXT NOT NULL,
    set_value TEXT,
    PRIMARY KEY ("set_name")
);
```

Configure Connection

Connection

Defines the connection that will be used to create the security module tables.

> All existing connections in the project will be displayed.

Mode

Use existing tables

When using this option, the developer must have the necessary tables in his database to create the security module.

Click here to check the tables necessary for this type of security.

Create tables

Once this option is selected, Scriptcase is responsible for creating the tables necessary to use the security module.

Table prefix

Defines a prefix for the tables that will be created by the security module.

> By default, Scriptcase uses **sec__**.

Delete tables if they already exist

This option eliminates tables with the same name in your database, based on the tables used by the module, listed below.

For example, by setting the prefix **prj__** to your tables, if there is any table with the name **prj_users** in your database, it will be deleted and a new table will be created.

> This option is only available when a **Create tables** option is selected.

Protect logged in users

This option prevents the same user from accessing the system simultaneously in different sessions. By checking this option, the **logged** table will be created.

> When using the default prefix sec_ the table name will be **sec_logged**.

Use social medias

This option allows the configuration of Facebook and Twitter to authenticate system users. When checking this option, the **users_social** table will be created.

> When using the default prefix sec_ the table name will be **sec_users_social**.

Table association

The **Associate Tables** screen will be displayed if the **Use existing tables** option is selected in the **Configure Connection** step.

> When using the **Create tables** option, this step will be carried out automatically, with the tables that will be created by the security module generator.

At this stage, fields from existing tables must be associated with security application fields (applications generated by the Security Module).

Check below the structure of the tables created for this type of security.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    "role" TEXT,
    "phone" TEXT,
    "pswd_last_updated" TIMESTAMP,
    "mfa_last_updated" TIMESTAMP DEFAULT NULL,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
    "app_name" TEXT NOT NULL,
    "app_type" TEXT,
    "description" TEXT,
    PRIMARY KEY ("app_name")
);
```

sec_groups

```
CREATE TABLE "sec_groups" (
    "group_id" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
    "description" TEXT
);
```

sec_users_groups

```
CREATE TABLE "sec_users_groups" (
    "login" TEXT NOT NULL,
    "group_id" INTEGER NOT NULL,
    PRIMARY KEY ("login", "group_id")
);
```

sec_groups_apps

```
CREATE TABLE "sec_groups_apps" (
    "group_id" INTEGER NOT NULL,
    "app_name" TEXT NOT NULL,
    "priv_access" TEXT,
    "priv_insert" TEXT,
    "priv_delete" TEXT,
    "priv_update" TEXT,
    "priv_export" TEXT,
    "priv_print" TEXT,
    PRIMARY KEY ("group_id", "app_name")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    login TEXT NOT NULL,
    date_login TEXT,
    sc_session TEXT,
    ip TEXT
);
```

> **NOTA:** A tabela **Usuários logados** será criada somente se a opção **Proteger usuários logados** for habilitada durante a criação do Módulo de Segurança.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> **NOTA:** A tabela **users_social** será criada somente se a opção **Utilizar redes sociais** for verificada durante a criação do Módulo de Segurança.

General

Prefix Application

Defines a prefix for the applications that will be created by the security module. By default, the field is filled with the prefix *app_*.

> This field is mandatory when creating the security module.

Folder

Defines the name of the folder that will be created in the project to group the applications created by the security generator. By default, the folder name is *Security*.

Enable Security

Defines whether the **Use Security** attribute will be enabled in the applications selected in the *Insert Applications* tab.

> By default, applications generated by the security module have the Use Security option enabled.

Remember login

Enables the **Remember me** option at system login.

Example of the Remember Me feature

This feature stores user data in cookies, and allows them to return to the system even after closing the session. If the user has logged out, the cookie data is deleted and he or she will have to log in again.

Cookie expiration time

Defines the lifetime, in days, of the cookie that contains the access data of users who selected the **Remember me** option.

> By default, Scriptcase sets the limit to 30 days.

When session expires

Determines system behavior when user session expires.

- **No action:** A message will be displayed, user not authorized, and an OK button to return to the login screen.
- **Redirect to login:** In this option the user will be redirected directly to the login screen.
- **Display a message and redirect to login:** An unauthorized user message will be displayed and you will be redirected to the login screen.

Theme

Defines the theme that will be used in applications created by the Security Module.

> To add or remove themes from the select, access the Project > Properties menu, there you can add or remove themes for the project.

Log

Defines the log module that will be used in the system. Click here and check out how to create a Log Module.

Menu

Defines the menu application used to create the security item, which contains the applications generated by the module for system management.

This attribute is mandatory for module creation. If the project already has a menu application, they will be listed and can be used in creation. The developer can choose to create a new menu application using the option: **Create application menu**.

Menu orientation

Defines the orientation of the menu that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Menu theme

Determines the initial theme of the menu application that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Use SweetAlert

Use SweetAlert to display application messages. When enabled, this option will override the browser’s “confirm” and “alert” settings.

Login

Here you can define the type and number of characters allowed for the username and password fields.

Label Position at Login

Defines the positioning of the fields' Label in relation to the data.

- **Above:** Positions the label above the Above input label at login.
- **Below:** Positions the label below the Down input label at login
- **Watermark:** Positions the label as a watermark.

Login mode

Defines how the user login will be performed.

- **User** - Default form of access, in this option

Cryptography

Use encryption to store the password in the user table. You have the following encryptions available: **MD5, SHA1, SHA256 and SHA512**

Show password characters

Defines whether the reveal password button will be displayed in the password field.

Password field with reveal password button

Use Captcha

It activates the captcha for the login application. The following options are available:

- **No:** Does not display the captcha in the login application.
- **Captcha:** Uses the built-in script library to display the captcha.
- **recaptcha:** Uses Google's recaptcha V2. To configure, see the video below:

  Watch a recaptcha setup video.

Secret Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Site Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Last password change

Defines the time, in days, until the system requests the user to change their password.

> If the value entered is 0, the password will not be reset.

Password strength

Defines the mandatory characters to compose the password for new users.

- **Capital letters**

- **Small letters**
- **Numbers**
- **Special characters**

Example of password strength in the application

Show language selection at login

Enables the display of the language field at login.

Example of language selection at login

Login Field

- **Minimum size**: Defines the minimum number of characters that can be used to define a login.
- **Maximum length**: Stipulates the maximum number of characters for defining a login.

Password field

- **Minimum length**: Minimum number of characters used to define the password.
- **Maximum length**: Defines the maximum number of characters for defining a password.

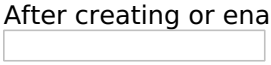
  The maximum values cannot be lower than the minimum values.

Login Template

When enabling the **Use a template attribute on the login screen**, the field for selecting the template that will be used will be available. Templates must be loaded/created in an external library, which in turn must be enabled for use in the project. See how to use an external library

> This option is only available when **Social Network Authentication** is not used.

If you do not have emplates created or enabled for the project, use the button to quickly access the external library settings.

After creating or enabling a template in the external library, if it is not listed, click on the button  to reload the select values.

E-mail Settings

API

This attribute will define whether an API or a customized SMTP will be used for the security module.

SMTP Server

Enter the SMTP server port. This information must be in accordance with the secure connection option.

Door

Defines the connection port according to the SSL used. Use 465 for **SSL**, 587 for **TLS**, or 25 for insecure connection. If you do not enter the port, Scriptcase applies the default port: 25.

Secure Connection:

Use **SSL** or **TSL**, or leave it blank for insecure connection.

SMTP User

Enter user information for SMTP connection. * **EX:** scriptcase@yahoo.com

SMTP Password

Enter the password for the SMTP connection.

SMTP email

Enter the outgoing SMTP email.

Sender's name

Sets the sender name

Email server

Select an email API to use in the security module or define a custom SMTP.

This email will be used in the options:

- All **Recover Password** options
- **New User** when selecting the options: **Requires activation by email**, **Send email to administrator**.

If you do not have an API created for use, check here how to create it.

Customized

Enter all SMTP settings required for operation.

> The options below are only available when the **Custom** option is selected.

- **The SMTP server:** Enter the SMTP server address.
  - **EX:** smtp.mail.yahoo.com
- **SMTP Port:** Enter the SMTP server port. This information must be in accordance with the secure connection option. Use 465 for **SSL**, 587 for **TLS**, or 25 for insecure connection. If you do not enter the port, Scriptcase applies the default port: 25.
- **Secure Connection:** Use **SSL** or **TSL**, or leave it blank for insecure connection.
- **SMTP User:** Enter user information for SMTP connection.
  - **EX:** scriptcase@yahoo.com

- **SMTP Password:** Enter the password for the SMTP connection.
- **SMTP email:** Enter the outgoing SMTP email.

SMTP

> To proceed with the password recovery, it is necessary that has the SMTP server configured for a successful process. Look here how to realize this procedure.

Yahoo

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.
> See also: Ways to securely access Yahoo Mail

SMTP Server Settings Yahoo	
SMTP Server	smtp.mail.yahoo.com
Port	465 or 587
E-mail address	Your full email address (name@domain.com.)
Password	The password for your account.
Requires SSL	Yes
Requires TSL	Yes (if available)
Requires authentication	Yes

Gmail

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.

SMTP Server Settings Gmail	
SMTP Server	smtp.gmail.com
E-mail address	Your full email address (Ex. example@gmail.com)
Password	The password for your account.
Port (TLS)	587
Port (SSL)	465
TLS / SSL Required	Yes

Outlook / Hotmail

For more information about these settings Click Here

SMTP Server Settings Outlook.com	
SMTP Server	smtp-mail.outlook.com
E-mail address	Your full email address (Ex. myname@outlook.com, not an alias)
Password	The password for your account.
Port	587 (you can use port 25 as an alternative)
TLS / SSL Required	Yes

New users

Settings for creating new users in the security system.

New Users

Allows new users to register

This option defines the availability of user registration directly in the login system. If you do not check this option, only users with administrative access can add new users.

Requires email activation

This option defines whether the new user must validate their registration by email to activate the account (Configure SMTP email to use this option)

Send email to administrator

This option defines whether the system administrator receives an email whenever a new user is registered. (Configure SMTP email to use this option)

Default Group

Defines the default group that the user will be inserted into when created.

> This feature is only available for the group security module or LDAP/Applications

Recover password

Send email with password

The system sends the user’s password via email only if SMTP has been configured correctly, and the password is not using encryption.

Reset password and send new one by email

The system automatically resets the password and sends it to the user via email (only if SMTP has been configured correctly).

Send email with a link to generate a new password

The system sends the user a link to access an application and reset the password.

> The email sending options, for example, *Send email with password* or *Requires activation by email* requires the correct configuration of the administrator’s email in the **Email Settings** tab.

Two Factor Authentication

Two-factor authentication (2FA)

Determines whether or not two-factor authentication is used in the system. Currently, the security module supports three ways of configuring resources:

- **Authentication**: This option will send the code to your Google authentication app.
- **SMS**: This option will send the authentication code via SMS.
- **Email**: This option will send the authentication code to the configured email.

Operation mode

In this attribute, the developer can define whether users will be required to use authentication to access the system or whether they may not use the resource.

- **All** - All users are required to configure
- **Individual** - Each user has the option of using authentication or not.

API

Lists the available APIs according to the type of authentication selected in the *Two-factor authentication (2FA)* attribute.

If you do not have any API created for the project, use the button to quickly access the API settings.

After configuring the API, click the to reload the select values.

> If you do not have an API created for use, check here how to create it.

Code expiration time

Set the time in **seconds** that the 2FA authentication token will expire.

,

Social Network

The Scriptcase allows user authentication through social networks such as **X** (formerly Twitter), **Facebook**, and **Google**. This feature makes application access easier by eliminating the need to create new credentials, making login more practical and secure.

To enable login with a social network, it is necessary to configure an application on the corresponding platform to obtain the access credentials.

> When using social network authentication, the **Login Template** option will not be available.

Facebook

Authentication via **Facebook** allows users to log in to the system using their social network accounts. To enable this feature, it is necessary to register an application on **Facebook for Developers**. See how to Generate Facebook credentials.

To use **Facebook**, check the **checkbox** to enable the **App ID** and **Secret** fields and enter the previously generated data.

App ID

Enter the **App ID** available in the settings of the application created in **Facebook Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **Facebook Developers**.

X (formerly Twitter)

Allows user authentication via **X** (formerly Twitter), using the platform's API. To configure this feature, it is necessary to create an application in the **X Developer Portal** to obtain the authentication credentials. See how to Generate X credentials.

To use login with **X**, check the **checkbox** to enable the **Key** and **Secret** fields and enter the previously generated data.

Key

Enter the **Key** available in the settings of the application created in **X Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **X Developers**.

Google

Login via **Google** allows users to authenticate using their platform accounts quickly and securely. To enable this feature in Scriptcase, it is necessary to configure an application in **Google Cloud Console**. See how to Generate Google credentials.

To use login with **Google**, check the **checkbox** to enable the **Client ID** and **Secret** fields and enter the previously generated data.

Client ID

Enter the **Client ID** available in the settings of the application created in **Google Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **Google Developers**.

Logged Users

This setting defines the system behavior for login protection.

This option is available if you checked **Protect logged in users** during the connection step.

Show logged in users

If you select this option, Scriptcase will also create a Grid Application with the Security Module to display a report with all users currently connected to the system.

Brute Force attack protection

Allows/disables blocking users after a few failed access attempts.

Brute Force block time (in minutes)

Time, in minutes, that the user will remain inaccessible after several failed access attempts. (Only available when Protection against Ruth Force Attacks is activated)

Number of attempts before blocking

Number of failed access attempts until protection is enabled. (Available only when protection is enabled for brute force attacks)

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the system and add new users.

Login

This option defines the system administrator user. By default, the user defined by Scriptcase is **admin**.

Password

This option sets the system administrator password. By default, the user defined by Scriptcase is **admin**.

> To increase the security of your project, we recommend changing the default username and password “admin” at this stage or after the first access.

Name

This option defines the name of the administrator user on the system.

Email

This option defines the email address of the system administrator user. This email can be used for some new user registration settings, such as notification to the administrator.

> For the email to be sent successfully to the administrator, the SMTP parameters must have been configured or an email API must have been selected in the previous step in Configurations.

Add Applications

This option adds the applications already created in the project to the Security Module applications table.

Include All

All project applications will be added to the application table.

Select Applications

Allows the developer to select specific applications to include in the application table. When clicking Select, all existing applications in the project will be listed.

Application selection screen

On this screen you can search for applications by name or filter by their types.

Once selected, the applications will be listed below.

List of selected applications

When unchecking this option, existing applications in the project when creating the module will need to be inserted through application synchronization, after executing the security module.

Save Profile

This feature will always be displayed when completing the security module configuration. The saved profiles will be displayed on the initial module creation screen, in the **Create / Choose Profile** step.

Viewing Saved Profiles

Ao clicar em **Usar**, o processo de criação do módulo de segurança se dará da mesma forma, porém, com as opções de configuração preenchidas com os valores salvos no perfil. When clicking on **Use**, the security module creation process will take place in the same way, however, with the configuration options filled in with the values saved in the profile.

Save Profile

When enabled, the settings made in the created module will be saved in the profile.

Name

When enabling profile saving, the name becomes a mandatory field to identify the created profile.

Level

Defines the viewing scope in which the profile will be saved.

- **Public:** Defines that the security profile will be available in any Scriptcase project.
- **Project:** Defines that the security profile will only be available in the project in which it was created.
- **User:** Defines that the security profile will only be available to the Scriptcase user who created it.

Building Security

Finally, a summary of the creation of the security module, listing the changes made by Scriptcase to your project.

- **Open Project:** You will be redirected to the home page of the project being modified.
- **Generate source code:** Generates the source code of all applications created by the security module.
- **Download SQL:** Makes the table structures of the created module available to the developer

  After creating the security module, it will be necessary to generate the source code of all applications that were marked by the security module usage flag. Except Login and Menu applications.

Application-Based Security Module

In the **User Security Module**, the system administrator has the ability to restrict access to different applications according to the user. In this type of security, the administrator must define which users can use certain applications, restricting access individually.

Check below the tables that will be created in your database when configuring a security module.

> The **sec_logged** (Logged User) and **sec_users_social** (Social Network Authentication) tables will be created if these features are enabled, the creation of the other tables is mandatory.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    "role" TEXT,
    "phone" TEXT,
    "pswd_last_updated" TIMESTAMP,
    "mfa_last_updated" TIMESTAMP DEFAULT NULL,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
    "app_name" TEXT NOT NULL,
    "app_type" TEXT,
    "description" TEXT,
    PRIMARY KEY ("app_name")
);
```

sec_users_apps

```
CREATE TABLE "sec_users_apps" (
"login" TEXT NOT NULL,
"app_name" TEXT NOT NULL,
"priv_access" TEXT,
"priv_insert" TEXT,
"priv_delete" TEXT,
"priv_update" TEXT,
"priv_export" TEXT,
"priv_print" TEXT,
PRIMARY KEY ("login", "app_name")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    login TEXT NOT NULL,
    date_login TEXT,
    sc_session TEXT,
    ip TEXT
);
```

> **NOTA:** A tabela **Usuários logados** será criada somente se a opção **Proteger usuários logados** for habilitada durante a criação do Módulo de Segurança.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
```

```
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> **NOTA:** A tabela **users_social** será criada somente se a opção **Utilizar redes sociais** for verificada durante a criação do Módulo de Segurança.

sec_settings

```
CREATE TABLE "sec_settings" (
    set_name TEXT NOT NULL,
    set_value TEXT,
    PRIMARY KEY ("set_name")
);
```

Configure Connection

Connection

Defines the connection that will be used to create the security module tables.

> All existing connections in the project will be displayed.

Mode

Use existing tables

When using this option, the developer must have the necessary tables in his database to create the security module.

Click here to check the tables necessary for this type of security.

Create tables

Once this option is selected, Scriptcase is responsible for creating the tables necessary to use the security module.

Table prefix

Defines a prefix for the tables that will be created by the security module.

> By default, Scriptcase uses **sec__**.

Delete tables if they already exist

This option eliminates tables with the same name in your database, based on the tables used by the module, listed below.

For example, by setting the prefix **prj__** to your tables, if there is any table with the name **prj_users** in your database, it will be deleted and a new table will be created.

> This option is only available when a **Create tables** option is selected.

Protect logged in users

This option prevents the same user from accessing the system simultaneously in different sessions. By checking this option, the **logged** table will be created.

> When using the default prefix sec_ the table name will be **sec_logged**.

Use social medias

This option allows the configuration of Facebook and Twitter to authenticate system users. When checking this option, the **users_social** table will be created.

> When using the default prefix sec_ the table name will be **sec_users_social**.

Association tables

The **Associate Tables** screen will be displayed if the **Use existing tables** option is selected in the **Configure Connection** step.

> When using the **Create tables** option, this step will be carried out automatically, with the tables that will be created by the security module generator.

At this stage, fields from existing tables must be associated with security application fields (applications generated by the Security Module).

Check below the structure of the tables created for this type of security.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    "role" TEXT,
    "phone" TEXT,
    "pswd_last_updated" TIMESTAMP,
    "mfa_last_updated" TIMESTAMP DEFAULT NULL,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
    "app_name" TEXT NOT NULL,
    "app_type" TEXT,
    "description" TEXT,
    PRIMARY KEY ("app_name")
);
```

sec_users_apps

```
CREATE TABLE "sec_users_apps" (
"login" TEXT NOT NULL,
"app_name" TEXT NOT NULL,
"priv_access" TEXT,
"priv_insert" TEXT,
"priv_delete" TEXT,
"priv_update" TEXT,
"priv_export" TEXT,
"priv_print" TEXT,
PRIMARY KEY ("login", "app_name")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    login TEXT NOT NULL,
    date_login TEXT,
    sc_session TEXT,
    ip TEXT
);
```

> **NOTA:** A tabela **Usuários logados** será criada somente se a opção **Proteger usuários logados** for habilitada durante a criação do Módulo de Segurança.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

NOTA: A tabela **users_social** será criada somente se a opção **Utilizar redes sociais** for verificada durante a criação do Módulo de Segurança.

General

Prefix Application

Defines a prefix for the applications that will be created by the security module. By default, the field is filled with the prefix *app_*.

> This field is mandatory when creating the security module.

Folder

Defines the name of the folder that will be created in the project to group the applications created by the security generator. By default, the folder name is *Security*.

Enable Security

Defines whether the **Use Security** attribute will be enabled in the applications selected in the *Insert Applications* tab.

> By default, applications generated by the security module have the Use Security option enabled.

Remember login

Enables the **Remember me** option at system login.

Example of the Remember Me feature

This feature stores user data in cookies, and allows them to return to the system even after closing the session. If the user has logged out, the cookie data is deleted and he or she will have to log in again.

Cookie expiration time

Defines the lifetime, in days, of the cookie that contains the access data of users who selected the **Remember me** option.

> By default, Scriptcase sets the limit to 30 days.

When session expires

Determines system behavior when user session expires.

- **No action:** A message will be displayed, user not authorized, and an OK button to return to the login screen.
- **Redirect to login:** In this option the user will be redirected directly to the login screen.
- **Display a message and redirect to login:** An unauthorized user message will be displayed and you will be redirected to the login screen.

Theme

Defines the theme that will be used in applications created by the Security Module.

> To add or remove themes from the select, access the Project > Properties menu, there you can add or remove themes for the project.

Log

Defines the log module that will be used in the system. Click here and check out how to create a Log Module.

Menu

Defines the menu application used to create the security item, which contains the applications generated by the module for system management.

This attribute is mandatory for module creation. If the project already has a menu application, they will be listed and can be used in creation. The developer can choose to create a new menu application using the option: **Create application menu**.

Menu orientation

Defines the orientation of the menu that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Menu theme

Determines the initial theme of the menu application that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Use SweetAlert

Use SweetAlert to display application messages. When enabled, this option will override the browser’s “confirm” and “alert” settings.

Login

Here you can define the type and number of characters allowed for the username and password fields.

Label Position at Login

Defines the positioning of the fields' Label in relation to the data.

- **Above:** Positions the label above the Above input label at login.
- **Below:** Positions the label below the Down input label at login
- **Watermark:** Positions the label as a watermark.

Login mode

Defines how the user login will be performed.

- **User** - Default form of access, in this option

Cryptography

Use encryption to store the password in the user table. You have the following encryptions available: **MD5, SHA1, SHA256 and SHA512**

Show password characters

Defines whether the reveal password button will be displayed in the password field.

Password field with reveal password button

Use Captcha

It activates the captcha for the login application. The following options are available:

- **No:** Does not display the captcha in the login application.
- **Captcha:** Uses the built-in script library to display the captcha.
- **recaptcha:** Uses Google's recaptcha V2. To configure, see the video below:

  Watch a recaptcha setup video.

Secret Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Site Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Last password change

Defines the time, in days, until the system requests the user to change their password.

> If the value entered is 0, the password will not be reset.

Password strength

Defines the mandatory characters to compose the password for new users.

- **Capital letters**

- **Small letters**
- **Numbers**
- **Special characters**

Example of password strength in the application

Show language selection at login

Enables the display of the language field at login.

Example of language selection at login

Login Field

- **Minimum size**: Defines the minimum number of characters that can be used to define a login.
- **Maximum length**: Stipulates the maximum number of characters for defining a login.

Password field

- **Minimum length**: Minimum number of characters used to define the password.
- **Maximum length**: Defines the maximum number of characters for defining a password.

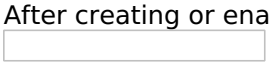
  The maximum values cannot be lower than the minimum values.

Login Template

When enabling the **Use a template attribute on the login screen**, the field for selecting the template that will be used will be available. Templates must be loaded/created in an external library, which in turn must be enabled for use in the project. See how to use an external library

> This option is only available when **Social Network Authentication** is not used.

If you do not have emplates created or enabled for the project, use the button to quickly access the external library settings.

After creating or enabling a template in the external library, if it is not listed, click on the button  to reload the select values.

E-mail Settings

API

This attribute will define whether an API or a customized SMTP will be used for the security module.

SMTP Server

Enter the SMTP server port. This information must be in accordance with the secure connection option.

Door

Defines the connection port according to the SSL used. Use 465 for **SSL**, 587 for **TLS**, or 25 for insecure connection. If you do not enter the port, Scriptcase applies the default port: 25.

Secure Connection:

Use **SSL** or **TSL**, or leave it blank for insecure connection.

SMTP User

Enter user information for SMTP connection. * **EX:** scriptcase@yahoo.com

SMTP Password

Enter the password for the SMTP connection.

SMTP email

Enter the outgoing SMTP email.

Sender's name

Sets the sender name

Email server

Select an email API to use in the security module or define a custom SMTP.

This email will be used in the options:

- All **Recover Password** options
- **New User** when selecting the options: **Requires activation by email**, **Send email to administrator**.

If you do not have an API created for use, check here how to create it.

Customized

Enter all SMTP settings required for operation.

> The options below are only available when the **Custom** option is selected.

- **The SMTP server:** Enter the SMTP server address.
  - **EX:** smtp.mail.yahoo.com
- **SMTP Port:** Enter the SMTP server port. This information must be in accordance with the secure connection option. Use 465 for **SSL**, 587 for **TLS**, or 25 for insecure connection. If you do not enter the port, Scriptcase applies the default port: 25.
- **Secure Connection:** Use **SSL** or **TSL**, or leave it blank for insecure connection.
- **SMTP User:** Enter user information for SMTP connection.
  - **EX:** scriptcase@yahoo.com

- **SMTP Password:** Enter the password for the SMTP connection.
- **SMTP email:** Enter the outgoing SMTP email.

SMTP

> To proceed with the password recovery, it is necessary that has the SMTP server configured for a successful process. Look here how to realize this procedure.

Yahoo

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.
> See also: Ways to securely access Yahoo Mail

SMTP Server Settings Yahoo	
SMTP Server	smtp.mail.yahoo.com
Port	465 or 587
E-mail address	Your full email address (name@domain.com.)
Password	The password for your account.
Requires SSL	Yes
Requires TSL	Yes (if available)
Requires authentication	Yes

Gmail

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.

SMTP Server Settings Gmail	
SMTP Server	smtp.gmail.com
E-mail address	Your full email address (Ex. example@gmail.com)
Password	The password for your account.
Port (TLS)	587
Port (SSL)	465
TLS / SSL Required	Yes

Outlook / Hotmail

For more information about these settings Click Here

SMTP Server Settings Outlook.com	
SMTP Server	smtp-mail.outlook.com
E-mail address	Your full email address (Ex. myname@outlook.com, not an alias)
Password	The password for your account.
Port	587 (you can use port 25 as an alternative)
TLS / SSL Required	Yes

New users

Settings for creating new users in the security system.

New Users

Allows new users to register

This option defines the availability of user registration directly in the login system. If you do not check this option, only users with administrative access can add new users.

Requires email activation

This option defines whether the new user must validate their registration by email to activate the account (Configure SMTP email to use this option)

Send email to administrator

This option defines whether the system administrator receives an email whenever a new user is registered. (Configure SMTP email to use this option)

Default Group

Defines the default group that the user will be inserted into when created.

> This feature is only available for the group security module or LDAP/Applications

Recover password

Send email with password

The system sends the user’s password via email only if SMTP has been configured correctly, and the password is not using encryption.

Reset password and send new one by email

The system automatically resets the password and sends it to the user via email (only if SMTP has been configured correctly).

Send email with a link to generate a new password

The system sends the user a link to access an application and reset the password.

> The email sending options, for example, *Send email with password* or *Requires activation by email* requires the correct configuration of the administrator’s email in the **Email Settings** tab.

Two Factor Authentication

Two-factor authentication (2FA)

Determines whether or not two-factor authentication is used in the system. Currently, the security module supports three ways of configuring resources:

- **Authentication**: This option will send the code to your Google authentication app.
- **SMS**: This option will send the authentication code via SMS.
- **Email**: This option will send the authentication code to the configured email.

Operation mode

In this attribute, the developer can define whether users will be required to use authentication to access the system or whether they may not use the resource.

- **All** - All users are required to configure
- **Individual** - Each user has the option of using authentication or not.

API

Lists the available APIs according to the type of authentication selected in the *Two-factor authentication (2FA)* attribute.

If you do not have any API created for the project, use the button to quickly access the API settings.

After configuring the API, click the to reload the select values.

> If you do not have an API created for use, check here how to create it.

Code expiration time

Set the time in **seconds** that the 2FA authentication token will expire.

,

Social Network

The Scriptcase allows user authentication through social networks such as **X** (formerly Twitter), **Facebook**, and **Google**. This feature makes application access easier by eliminating the need to create new credentials, making login more practical and secure.

To enable login with a social network, it is necessary to configure an application on the corresponding platform to obtain the access credentials.

> When using social network authentication, the **Login Template** option will not be available.

Facebook

Authentication via **Facebook** allows users to log in to the system using their social network accounts. To enable this feature, it is necessary to register an application on **Facebook for Developers**. See how to Generate Facebook credentials.

To use **Facebook**, check the **checkbox** to enable the **App ID** and **Secret** fields and enter the previously generated data.

App ID

Enter the **App ID** available in the settings of the application created in **Facebook Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **Facebook Developers**.

X (formerly Twitter)

Allows user authentication via **X** (formerly Twitter), using the platform's API. To configure this feature, it is necessary to create an application in the **X Developer Portal** to obtain the authentication credentials. See how to Generate X credentials.

To use login with **X**, check the **checkbox** to enable the **Key** and **Secret** fields and enter the previously generated data.

Key

Enter the **Key** available in the settings of the application created in **X Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **X Developers**.

Google

Login via **Google** allows users to authenticate using their platform accounts quickly and securely. To enable this feature in Scriptcase, it is necessary to configure an application in **Google Cloud Console**. See how to Generate Google credentials.

To use login with **Google**, check the **checkbox** to enable the **Client ID** and **Secret** fields and enter the previously generated data.

Client ID

Enter the **Client ID** available in the settings of the application created in **Google Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **Google Developers**.

Logged Users

This setting defines the system behavior for login protection.

This option is available if you checked **Protect logged in users** during the connection step.

Show logged in users

If you select this option, Scriptcase will also create a Grid Application with the Security Module to display a report with all users currently connected to the system.

Brute Force attack protection

Allows/disables blocking users after a few failed access attempts.

Brute Force block time (in minutes)

Time, in minutes, that the user will remain inaccessible after several failed access attempts. (Only available when Protection against Ruth Force Attacks is activated)

Number of attempts before blocking

Number of failed access attempts until protection is enabled. (Available only when protection is enabled for brute force attacks)

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the system and add new users.

Login

This option defines the system administrator user. By default, the user defined by Scriptcase is **admin**.

Password

This option sets the system administrator password. By default, the user defined by Scriptcase is **admin**.

> To increase the security of your project, we recommend changing the default username and password “admin” at this stage or after the first access.

Name

This option defines the name of the administrator user on the system.

Email

This option defines the email address of the system administrator user. This email can be used for some new user registration settings, such as notification to the administrator.

> For the email to be sent successfully to the administrator, the SMTP parameters must have been configured or an email API must have been selected in the previous step in Configurations.

Add Applications

This option adds the applications already created in the project to the Security Module applications table.

Include All

All project applications will be added to the application table.

Select Applications

Allows the developer to select specific applications to include in the application table. When clicking Select, all existing applications in the project will be listed.

Application selection screen

On this screen you can search for applications by name or filter by their types.

Once selected, the applications will be listed below.

List of selected applications

When unchecking this option, existing applications in the project when creating the module will need to be inserted through application synchronization, after executing the security module.

Save Profile

This feature will always be displayed when completing the security module configuration. The saved profiles will be displayed on the initial module creation screen, in the **Create / Choose Profile** step.

Viewing Saved Profiles

Ao clicar em **Usar**, o processo de criação do módulo de segurança se dará da mesma forma, porém, com as opções de configuração preenchidas com os valores salvos no perfil. When clicking on **Use**, the security module creation process will take place in the same way, however, with the configuration options filled in with the values saved in the profile.

Save Profile

When enabled, the settings made in the created module will be saved in the profile.

Name

When enabling profile saving, the name becomes a mandatory field to identify the created profile.

Level

Defines the viewing scope in which the profile will be saved.

- **Public:** Defines that the security profile will be available in any Scriptcase project.
- **Project:** Defines that the security profile will only be available in the project in which it was created.
- **User:** Defines that the security profile will only be available to the Scriptcase user who created it.

Building Security

Finally, a summary of the creation of the security module, listing the changes made by Scriptcase to your project.

- **Open Project:** You will be redirected to the home page of the project being modified.
- **Generate source code:** Generates the source code of all applications created by the security module.
- **Download SQL:** Makes the table structures of the created module available to the developer

  After creating the security module, it will be necessary to generate the source code of all applications that were marked by the security module usage flag. Except Login and Menu applications.

User-Based Security Module

In the **Security Module by User**, all registered users have access to the system applications, functioning only as authentication for each user.

Check below the tables that will be created in your database when configuring a security module.

> The **sec_logged** (Logged User) and **sec_users_social** (Social Network Authentication) tables will be created if these features are enabled, the creation of the other tables is mandatory.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    "role" TEXT,
    "phone" TEXT,
    "pswd_last_updated" TIMESTAMP,
    "mfa_last_updated" TIMESTAMP DEFAULT NULL,
    PRIMARY KEY ("login")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    login TEXT NOT NULL,
    date_login TEXT,
    sc_session TEXT,
    ip TEXT
);
```

> **NOTA:** A tabela **Usuários logados** será criada somente se a opção **Proteger usuários logados** for habilitada durante a criação do Módulo de Segurança.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> **NOTA:** A tabela **users_social** será criada somente se a opção **Utilizar redes sociais** for verificada durante a criação do Módulo de Segurança.

sec_settings

```
CREATE TABLE "sec_settings" (
    set_name TEXT NOT NULL,
    set_value TEXT,
    PRIMARY KEY ("set_name")
);
```

Configure Connection

Connection

Defines the connection that will be used to create the security module tables.

> All existing connections in the project will be displayed.

Mode

Use existing tables

When using this option, the developer must have the necessary tables in his database to create the security module.

Click here to check the tables necessary for this type of security.

Create tables

Once this option is selected, Scriptcase is responsible for creating the tables necessary to use the security module.

Table prefix

Defines a prefix for the tables that will be created by the security module.

> By default, Scriptcase uses **sec__**.

Delete tables if they already exist

This option eliminates tables with the same name in your database, based on the tables used by the module, listed below.

For example, by setting the prefix **prj__** to your tables, if there is any table with the name **prj_users** in your database, it will be deleted and a new table will be created.

> This option is only available when a **Create tables** option is selected.

Protect logged in users

This option prevents the same user from accessing the system simultaneously in different sessions. By checking this option, the **logged** table will be created.

> When using the default prefix sec_ the table name will be **sec_logged**.

Use social medias

This option allows the configuration of Facebook and Twitter to authenticate system users. When checking this option, the **users_social** table will be created.

> When using the default prefix sec_ the table name will be **sec_users_social**.

Association tables

The **Associate Tables** screen will be displayed if the **Use existing tables** option is selected in the **Configure Connection** step.

> When using the **Create tables** option, this step will be carried out automatically, with the tables that will be created by the security module generator.

At this stage, fields from existing tables must be associated with security application fields (applications generated by the Security Module).

Check below the structure of the tables created for this type of security.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    "role" TEXT,
    "phone" TEXT,
    "pswd_last_updated" TIMESTAMP,
    "mfa_last_updated" TIMESTAMP DEFAULT NULL,
    PRIMARY KEY ("login")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    login TEXT NOT NULL,
    date_login TEXT,
    sc_session TEXT,
    ip TEXT
);
```

> **NOTA:** A tabela **Usuários logados** será criada somente se a opção **Proteger usuários logados** for habilitada durante a criação do Módulo de Segurança.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> **NOTA:** A tabela **users_social** será criada somente se a opção **Utilizar redes sociais** for verificada durante a criação do Módulo de Segurança.

General

Prefix Application

Defines a prefix for the applications that will be created by the security module. By default, the field is filled with the prefix *app_*.

> This field is mandatory when creating the security module.

Folder

Defines the name of the folder that will be created in the project to group the applications created by the security generator. By default, the folder name is *Security*.

Enable Security

Defines whether the **Use Security** attribute will be enabled in the applications selected in the *Insert Applications* tab.

> By default, applications generated by the security module have the Use Security option enabled.

Remember login

Enables the **Remember me** option at system login.

Example of the Remember Me feature

This feature stores user data in cookies, and allows them to return to the system even after closing the session. If the user has logged out, the cookie data is deleted and he or she will have to log in again.

Cookie expiration time

Defines the lifetime, in days, of the cookie that contains the access data of users who selected the **Remember me** option.

> By default, Scriptcase sets the limit to 30 days.

When session expires

Determines system behavior when user session expires.

- **No action:** A message will be displayed, user not authorized, and an OK button to return to the login screen.
- **Redirect to login:** In this option the user will be redirected directly to the login screen.
- **Display a message and redirect to login:** An unauthorized user message will be displayed and you will be redirected to the login screen.

Theme

Defines the theme that will be used in applications created by the Security Module.

> To add or remove themes from the select, access the Project > Properties menu, there you can add or remove themes for the project.

Log

Defines the log module that will be used in the system. Click here and check out how to create a Log Module.

Menu

Defines the menu application used to create the security item, which contains the applications generated by the module for system management.

This attribute is mandatory for module creation. If the project already has a menu application, they will be listed and can be used in creation. The developer can choose to create a new menu application using the option: **Create application menu**.

Menu orientation

Defines the orientation of the menu that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Menu theme

Determines the initial theme of the menu application that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Use SweetAlert

Use SweetAlert to display application messages. When enabled, this option will override the browser’s “confirm” and “alert” settings.

Login

Here you can define the type and number of characters allowed for the username and password fields.

Label Position at Login

Defines the positioning of the fields' Label in relation to the data.

- **Above:** Positions the label above the Above input label at login.
- **Below:** Positions the label below the Down input label at login
- **Watermark:** Positions the label as a watermark.

Login mode

Defines how the user login will be performed.

- **User** - Default form of access, in this option

Cryptography

Use encryption to store the password in the user table. You have the following encryptions available: **MD5, SHA1, SHA256 and SHA512**

Show password characters

Defines whether the reveal password button will be displayed in the password field.

Password field with reveal password button

Use Captcha

It activates the captcha for the login application. The following options are available:

- **No:** Does not display the captcha in the login application.
- **Captcha:** Uses the built-in script library to display the captcha.
- **recaptcha:** Uses Google's recaptcha V2. To configure, see the video below:

  Watch a recaptcha setup video.

Secret Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Site Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Last password change

Defines the time, in days, until the system requests the user to change their password.

> If the value entered is 0, the password will not be reset.

Password strength

Defines the mandatory characters to compose the password for new users.

- **Capital letters**

- **Small letters**
- **Numbers**
- **Special characters**

Example of password strength in the application

Show language selection at login

Enables the display of the language field at login.

Example of language selection at login

Login Field

- **Minimum size**: Defines the minimum number of characters that can be used to define a login.
- **Maximum length**: Stipulates the maximum number of characters for defining a login.

Password field

- **Minimum length**: Minimum number of characters used to define the password.
- **Maximum length**: Defines the maximum number of characters for defining a password.

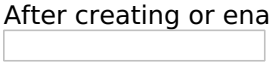
  The maximum values cannot be lower than the minimum values.

Login Template

When enabling the **Use a template attribute on the login screen**, the field for selecting the template that will be used will be available. Templates must be loaded/created in an external library, which in turn must be enabled for use in the project. See how to use an external library

> This option is only available when **Social Network Authentication** is not used.

If you do not have emplates created or enabled for the project, use the button to quickly access the external library settings.

After creating or enabling a template in the external library, if it is not listed, click on the button  to reload the select values.

E-mail Settings

API

This attribute will define whether an API or a customized SMTP will be used for the security module.

SMTP Server

Enter the SMTP server port. This information must be in accordance with the secure connection option.

Door

Defines the connection port according to the SSL used. Use 465 for **SSL**, 587 for **TLS**, or 25 for insecure connection. If you do not enter the port, Scriptcase applies the default port: 25.

Secure Connection:

Use **SSL** or **TSL**, or leave it blank for insecure connection.

SMTP User

Enter user information for SMTP connection. * **EX:** scriptcase@yahoo.com

SMTP Password

Enter the password for the SMTP connection.

SMTP email

Enter the outgoing SMTP email.

Sender's name

Sets the sender name

Email server

Select an email API to use in the security module or define a custom SMTP.

This email will be used in the options:

- All **Recover Password** options
- **New User** when selecting the options: **Requires activation by email**, **Send email to administrator**.

If you do not have an API created for use, check here how to create it.

Customized

Enter all SMTP settings required for operation.

> The options below are only available when the **Custom** option is selected.

- **The SMTP server:** Enter the SMTP server address.
  - **EX:** smtp.mail.yahoo.com
- **SMTP Port:** Enter the SMTP server port. This information must be in accordance with the secure connection option. Use 465 for **SSL**, 587 for **TLS**, or 25 for insecure connection. If you do not enter the port, Scriptcase applies the default port: 25.
- **Secure Connection:** Use **SSL** or **TSL**, or leave it blank for insecure connection.
- **SMTP User:** Enter user information for SMTP connection.
  - **EX:** scriptcase@yahoo.com

- **SMTP Password:** Enter the password for the SMTP connection.
- **SMTP email:** Enter the outgoing SMTP email.

SMTP

> To proceed with the password recovery, it is necessary that has the SMTP server configured for a successful process. Look here how to realize this procedure.

Yahoo

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.
> See also: Ways to securely access Yahoo Mail

SMTP Server Settings Yahoo	
SMTP Server	smtp.mail.yahoo.com
Port	465 or 587
E-mail address	Your full email address (name@domain.com.)
Password	The password for your account.
Requires SSL	Yes
Requires TSL	Yes (if available)
Requires authentication	Yes

Gmail

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.

SMTP Server Settings Gmail	
SMTP Server	smtp.gmail.com
E-mail address	Your full email address (Ex. example@gmail.com)
Password	The password for your account.
Port (TLS)	587
Port (SSL)	465
TLS / SSL Required	Yes

Outlook / Hotmail

For more information about these settings Click Here

SMTP Server Settings Outlook.com	
SMTP Server	smtp-mail.outlook.com
E-mail address	Your full email address (Ex. myname@outlook.com, not an alias)
Password	The password for your account.
Port	587 (you can use port 25 as an alternative)
TLS / SSL Required	Yes

New users

Settings for creating new users in the security system.

New Users

Allows new users to register

This option defines the availability of user registration directly in the login system. If you do not check this option, only users with administrative access can add new users.

Requires email activation

This option defines whether the new user must validate their registration by email to activate the account (Configure SMTP email to use this option)

Send email to administrator

This option defines whether the system administrator receives an email whenever a new user is registered. (Configure SMTP email to use this option)

Default Group

Defines the default group that the user will be inserted into when created.

> This feature is only available for the group security module or LDAP/Applications

Recover password

Send email with password

The system sends the user’s password via email only if SMTP has been configured correctly, and the password is not using encryption.

Reset password and send new one by email

The system automatically resets the password and sends it to the user via email (only if SMTP has been configured correctly).

Send email with a link to generate a new password

The system sends the user a link to access an application and reset the password.

> The email sending options, for example, *Send email with password* or *Requires activation by email* requires the correct configuration of the administrator’s email in the **Email Settings** tab.

Two Factor Authentication

Two-factor authentication (2FA)

Determines whether or not two-factor authentication is used in the system. Currently, the security module supports three ways of configuring resources:

- **Authentication**: This option will send the code to your Google authentication app.
- **SMS**: This option will send the authentication code via SMS.
- **Email**: This option will send the authentication code to the configured email.

Operation mode

In this attribute, the developer can define whether users will be required to use authentication to access the system or whether they may not use the resource.

- **All** - All users are required to configure
- **Individual** - Each user has the option of using authentication or not.

API

Lists the available APIs according to the type of authentication selected in the *Two-factor authentication (2FA)* attribute.

If you do not have any API created for the project, use the button to quickly access the API settings.

After configuring the API, click the to reload the select values.

> If you do not have an API created for use, check here how to create it.

Code expiration time

Set the time in **seconds** that the 2FA authentication token will expire.

,

Social Network

The Scriptcase allows user authentication through social networks such as **X** (formerly Twitter), **Facebook**, and **Google**. This feature makes application access easier by eliminating the need to create new credentials, making login more practical and secure.

To enable login with a social network, it is necessary to configure an application on the corresponding platform to obtain the access credentials.

> When using social network authentication, the **Login Template** option will not be available.

Facebook

Authentication via **Facebook** allows users to log in to the system using their social network accounts. To enable this feature, it is necessary to register an application on **Facebook for Developers**. See how to Generate Facebook credentials.

To use **Facebook**, check the **checkbox** to enable the **App ID** and **Secret** fields and enter the previously generated data.

App ID

Enter the **App ID** available in the settings of the application created in **Facebook Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **Facebook Developers**.

X (formerly Twitter)

Allows user authentication via **X** (formerly Twitter), using the platform's API. To configure this feature, it is necessary to create an application in the **X Developer Portal** to obtain the authentication credentials. See how to Generate X credentials.

To use login with **X**, check the **checkbox** to enable the **Key** and **Secret** fields and enter the previously generated data.

Key

Enter the **Key** available in the settings of the application created in **X Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **X Developers**.

Google

Login via **Google** allows users to authenticate using their platform accounts quickly and securely. To enable this feature in Scriptcase, it is necessary to configure an application in **Google Cloud Console**. See how to Generate Google credentials.

To use login with **Google**, check the **checkbox** to enable the **Client ID** and **Secret** fields and enter the previously generated data.

Client ID

Enter the **Client ID** available in the settings of the application created in **Google Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **Google Developers**.

Logged Users

This setting defines the system behavior for login protection.

This option is available if you checked **Protect logged in users** during the connection step.

Show logged in users

If you select this option, Scriptcase will also create a Grid Application with the Security Module to display a report with all users currently connected to the system.

Brute Force attack protection

Allows/disables blocking users after a few failed access attempts.

Brute Force block time (in minutes)

Time, in minutes, that the user will remain inaccessible after several failed access attempts. (Only available when Protection against Ruth Force Attacks is activated)

Number of attempts before blocking

Number of failed access attempts until protection is enabled. (Available only when protection is enabled for brute force attacks)

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the system and add new users.

Login

This option defines the system administrator user. By default, the user defined by Scriptcase is **admin**.

Password

This option sets the system administrator password. By default, the user defined by Scriptcase is **admin**.

> To increase the security of your project, we recommend changing the default username and password “admin” at this stage or after the first access.

Name

This option defines the name of the administrator user on the system.

Email

This option defines the email address of the system administrator user. This email can be used for some new user registration settings, such as notification to the administrator.

> For the email to be sent successfully to the administrator, the SMTP parameters must have been configured or an email API must have been selected in the previous step in Configurations.

Add Applications

This option adds the applications already created in the project to the Security Module applications table.

Include All

All project applications will be added to the application table.

Select Applications

Allows the developer to select specific applications to include in the application table. When clicking Select, all existing applications in the project will be listed.

Application selection screen

On this screen you can search for applications by name or filter by their types.

Once selected, the applications will be listed below.

List of selected applications

When unchecking this option, existing applications in the project when creating the module will need to be inserted through application synchronization, after executing the security module.

Save Profile

This feature will always be displayed when completing the security module configuration. The saved profiles will be displayed on the initial module creation screen, in the **Create / Choose Profile** step.

Viewing Saved Profiles

Ao clicar em **Usar**, o processo de criação do módulo de segurança se dará da mesma forma, porém, com as opções de configuração preenchidas com os valores salvos no perfil. When clicking on **Use**, the security module creation process will take place in the same way, however, with the configuration options filled in with the values saved in the profile.

Save Profile

When enabled, the settings made in the created module will be saved in the profile.

Name

When enabling profile saving, the name becomes a mandatory field to identify the created profile.

Level

Defines the viewing scope in which the profile will be saved.

- **Public:** Defines that the security profile will be available in any Scriptcase project.
- **Project:** Defines that the security profile will only be available in the project in which it was created.
- **User:** Defines that the security profile will only be available to the Scriptcase user who created it.

Building Security

Finally, a summary of the creation of the security module, listing the changes made by Scriptcase to your project.

- **Open Project:** You will be redirected to the home page of the project being modified.
- **Generate source code:** Generates the source code of all applications created by the security module.
- **Download SQL:** Makes the table structures of the created module available to the developer

  After creating the security module, it will be necessary to generate the source code of all applications that were marked by the security module usage flag. Except Login and Menu applications.

LDAP Security Module

The **LDAP security module** works in a similar way to the per-user module, where everyone has access to the system applications, functioning only as user authentication.

This type of security does not create any table, with all user access control being carried out by the Ldap server.

Use Existing Tables

Connection

Defines the connection that will be used in security module applications.

> This type of security does not generate any tables, as all control is performed through Ldap

General

Prefix Application

Defines a prefix for the applications that will be created by the security module. By default, the field is filled with the prefix *app_*.

> This field is mandatory when creating the security module.

Folder

Defines the name of the folder that will be created in the project to group the applications created by the security generator. By default, the folder name is *Security*.

Enable Security

Defines whether the **Use Security** attribute will be enabled in the applications selected in the *Insert Applications* tab.

> By default, applications generated by the security module have the Use Security option enabled.

Remember login

Enables the **Remember me** option at system login.

Example of the Remember Me feature

This feature stores user data in cookies, and allows them to return to the system even after closing the session. If the user has logged out, the cookie data is deleted and he or she will have to log in again.

Cookie expiration time

Defines the lifetime, in days, of the cookie that contains the access data of users who selected the **Remember me** option.

> By default, Scriptcase sets the limit to 30 days.

When session expires

Determines system behavior when user session expires.

- **No action:** A message will be displayed, user not authorized, and an OK button to return to the login screen.
- **Redirect to login:** In this option the user will be redirected directly to the login screen.
- **Display a message and redirect to login:** An unauthorized user message will be displayed and you will be redirected to the login screen.

Theme

Defines the theme that will be used in applications created by the Security Module.

> To add or remove themes from the select, access the Project > Properties menu, there you can add or remove themes for the project.

Log

Defines the log module that will be used in the system. Click here and check out how to create a Log Module.

Menu

Defines the menu application used to create the security item, which contains the applications generated by the module for system management.

This attribute is mandatory for module creation. If the project already has a menu application, they will be listed and can be used in creation. The developer can choose to create a new menu application using the option: **Create application menu**.

Menu orientation

Defines the orientation of the menu that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Menu theme

Determines the initial theme of the menu application that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Use SweetAlert

Use SweetAlert to display application messages. When enabled, this option will override the browser’s “confirm” and “alert” settings.

Login

Here you can define the type and number of characters allowed for the username and password fields.

Label Position at Login

Defines the positioning of the fields' Label in relation to the data.

- **Above:** Positions the label above the Above input label at login.
- **Below:** Positions the label below the Down input label at login
- **Watermark:** Positions the label as a watermark.

Login mode

Defines how the user login will be performed.

- **User** - Default form of access, in this option

Cryptography

Use encryption to store the password in the user table. You have the following encryptions available: **MD5, SHA1, SHA256 and SHA512**

Show password characters

Defines whether the reveal password button will be displayed in the password field.

Password field with reveal password button

Use Captcha

It activates the captcha for the login application. The following options are available:

- **No:** Does not display the captcha in the login application.
- **Captcha:** Uses the built-in script library to display the captcha.
- **recaptcha:** Uses Google's recaptcha V2. To configure, see the video below:

  Watch a recaptcha setup video.

Secret Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Site Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Last password change

Defines the time, in days, until the system requests the user to change their password.

> If the value entered is 0, the password will not be reset.

Password strength

Defines the mandatory characters to compose the password for new users.

- **Capital letters**

- **Small letters**
- **Numbers**
- **Special characters**

Example of password strength in the application

Show language selection at login

Enables the display of the language field at login.

Example of language selection at login

Login Field

- **Minimum size**: Defines the minimum number of characters that can be used to define a login.
- **Maximum length**: Stipulates the maximum number of characters for defining a login.

Password field

- **Minimum length**: Minimum number of characters used to define the password.
- **Maximum length**: Defines the maximum number of characters for defining a password.

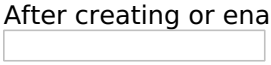
  The maximum values cannot be lower than the minimum values.

Login Template

When enabling the **Use a template attribute on the login screen**, the field for selecting the template that will be used will be available. Templates must be loaded/created in an external library, which in turn must be enabled for use in the project. See how to use an external library

> This option is only available when **Social Network Authentication** is not used.

If you do not have emplates created or enabled for the project, use the button to quickly access the external library settings.

After creating or enabling a template in the external library, if it is not listed, click on the button  to reload the select values.

Ldap

In the LDAP configurations, it is necessary to enter the access data for the previously configured server.

Server

Enter the LDAP server IP here.

DN

Enter the input attributes.

Door

Enter the server access door here. **The default port is 389.**

Suffix

Enter the user suffix.

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the system and add new users.

Login

This option defines the system administrator user. By default, the user defined by Scriptcase is **admin**.

Password

This option sets the system administrator password. By default, the user defined by Scriptcase is **admin**.

> To increase the security of your project, we recommend changing the default username and password “admin” at this stage or after the first access.

Name

This option defines the name of the administrator user on the system.

Email

This option defines the email address of the system administrator user. This email can be used for some new user registration settings, such as notification to the administrator.

> For the email to be sent successfully to the administrator, the SMTP parameters must have been configured or an email API must have been selected in the previous step in Configurations.

Add Applications

This option adds the applications already created in the project to the Security Module applications table.

Include All

All project applications will be added to the application table.

Select Applications

Allows the developer to select specific applications to include in the application table. When clicking Select, all existing applications in the project will be listed.

Application selection screen

On this screen you can search for applications by name or filter by their types.

Once selected, the applications will be listed below.

List of selected applications

When unchecking this option, existing applications in the project when creating the module will need to be inserted through application synchronization, after executing the security module.

Save Profile

This feature will always be displayed when completing the security module configuration. The saved profiles will be displayed on the initial module creation screen, in the **Create / Choose Profile** step.

Viewing Saved Profiles

Ao clicar em **Usar**, o processo de criação do módulo de segurança se dará da mesma forma, porém, com as opções de configuração preenchidas com os valores salvos no perfil. When clicking on **Use**, the security module creation process will take place in the same way, however, with the configuration options filled in with the values saved in the profile.

Save Profile

When enabled, the settings made in the created module will be saved in the profile.

Name

When enabling profile saving, the name becomes a mandatory field to identify the created profile.

Level

Defines the viewing scope in which the profile will be saved.

- **Public:** Defines that the security profile will be available in any Scriptcase project.
- **Project:** Defines that the security profile will only be available in the project in which it was created.
- **User:** Defines that the security profile will only be available to the Scriptcase user who created it.

Building Security

Finally, a summary of the creation of the security module, listing the changes made by Scriptcase to your project.

- **Open Project:** You will be redirected to the home page of the project being modified.
- **Generate source code:** Generates the source code of all applications created by the security module.
- **Download SQL:** Makes the table structures of the created module available to the developer

  After creating the security module, it will be necessary to generate the source code of all applications that were marked by the security module usage flag. Except Login and Menu applications.

LDAP-Based Application Security Module

In **LDAP/Application Security Module** it works in a similar way to the group module, where the system administrator has the ability to restrict access to different applications based on predefined user groups. This means that each group can have specific permissions to access certain functionalities or areas of the system, offering precise control over the actions allowed for each set of users.

Check below the tables that will be created in your database when configuring a security module.

> The **sec_logged** (Logged User) and **sec_users_social** (Social Network Authentication) tables will be created if these features are enabled, the creation of the other tables is mandatory.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    "role" TEXT,
    "phone" TEXT,
    "pswd_last_updated" TIMESTAMP,
    "mfa_last_updated" TIMESTAMP DEFAULT NULL,
    PRIMARY KEY ("login")
);
```

sec_apps

```
CREATE TABLE "sec_apps" (
    "app_name" TEXT NOT NULL,
    "app_type" TEXT,
    "description" TEXT,
    PRIMARY KEY ("app_name")
);
```

sec_users_apps

```
CREATE TABLE "sec_users_apps" (
"login" TEXT NOT NULL,
"app_name" TEXT NOT NULL,
"priv_access" TEXT,
"priv_insert" TEXT,
"priv_delete" TEXT,
"priv_update" TEXT,
"priv_export" TEXT,
"priv_print" TEXT,
PRIMARY KEY ("login", "app_name")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    login TEXT NOT NULL,
    date_login TEXT,
    sc_session TEXT,
    ip TEXT
);
```

> **NOTA:** A tabela **Usuários logados** será criada somente se a opção **Proteger usuários logados** for habilitada durante a criação do Módulo de Segurança.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
```

```
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> **NOTA:** A tabela **users_social** será criada somente se a opção **Utilizar redes sociais** for verificada durante a criação do Módulo de Segurança.

sec_settings

```
CREATE TABLE "sec_settings" (
    set_name TEXT NOT NULL,
    set_value TEXT,
    PRIMARY KEY ("set_name")
);
```

Configure Connection

Connection

Defines the connection that will be used to create the security module tables.

> All existing connections in the project will be displayed.

Mode

Use existing tables

When using this option, the developer must have the necessary tables in his database to create the security module.

Click here to check the tables necessary for this type of security.

Create tables

Once this option is selected, Scriptcase is responsible for creating the tables necessary to use the security module.

Table prefix

Defines a prefix for the tables that will be created by the security module.

> By default, Scriptcase uses **sec__**.

Delete tables if they already exist

This option eliminates tables with the same name in your database, based on the tables used by the module, listed below.

For example, by setting the prefix **prj__** to your tables, if there is any table with the name **prj_users** in your database, it will be deleted and a new table will be created.

> This option is only available when a **Create tables** option is selected.

Protect logged in users

This option prevents the same user from accessing the system simultaneously in different sessions. By checking this option, the **logged** table will be created.

> When using the default prefix sec_ the table name will be **sec_logged**.

Use social medias

This option allows the configuration of Facebook and Twitter to authenticate system users. When checking this option, the **users_social** table will be created.

> When using the default prefix sec_ the table name will be **sec_users_social**.

Association tables

The **Associate Tables** screen will be displayed if the **Use existing tables** option is selected in the **Configure Connection** step.

> When using the **Create tables** option, this step will be carried out automatically, with the tables that will be created by the security module generator.

At this stage, fields from existing tables must be associated with security application fields (applications generated by the Security Module).

Check below the structure of the tables created for this type of security.

sec_users

```
CREATE TABLE "sec_users" (
    "login" TEXT NOT NULL,
    "pswd" TEXT NOT NULL,
    "name" TEXT,
    "email" TEXT,
    "active" TEXT,
    "activation_code" TEXT,
    "priv_admin" TEXT,
    "mfa" TEXT,
    "picture" BLOB,
    "role" TEXT,
    "phone" TEXT,
    "pswd_last_updated" TIMESTAMP,
    "mfa_last_updated" TIMESTAMP DEFAULT NULL,
    PRIMARY KEY ("login")
);
```

sec_logged

```
CREATE TABLE "sec_logged" (
    login TEXT NOT NULL,
    date_login TEXT,
    sc_session TEXT,
    ip TEXT
);
```

> **NOTA:** A tabela **Usuários logados** será criada somente se a opção **Proteger usuários logados** for habilitada durante a criação do Módulo de Segurança.

sec_users_social

```
CREATE TABLE "sec_users_social" (
  "login" TEXT NOT NULL,
  "resource" TEXT NOT NULL,
  "resource_id" TEXT NOT NULL,
  PRIMARY KEY ("login", "resource", "resource_id")
);
```

> **NOTA:** A tabela **users_social** será criada somente se a opção **Utilizar redes sociais** for verificada durante a criação do Módulo de Segurança.

General

Prefix Application

Defines a prefix for the applications that will be created by the security module. By default, the field is filled with the prefix *app_*.

> This field is mandatory when creating the security module.

Folder

Defines the name of the folder that will be created in the project to group the applications created by the security generator. By default, the folder name is *Security*.

Enable Security

Defines whether the **Use Security** attribute will be enabled in the applications selected in the *Insert Applications* tab.

> By default, applications generated by the security module have the Use Security option enabled.

Remember login

Enables the **Remember me** option at system login.

Example of the Remember Me feature

This feature stores user data in cookies, and allows them to return to the system even after closing the session. If the user has logged out, the cookie data is deleted and he or she will have to log in again.

Cookie expiration time

Defines the lifetime, in days, of the cookie that contains the access data of users who selected the **Remember me** option.

> By default, Scriptcase sets the limit to 30 days.

When session expires

Determines system behavior when user session expires.

- **No action:** A message will be displayed, user not authorized, and an OK button to return to the login screen.
- **Redirect to login:** In this option the user will be redirected directly to the login screen.
- **Display a message and redirect to login:** An unauthorized user message will be displayed and you will be redirected to the login screen.

Theme

Defines the theme that will be used in applications created by the Security Module.

> To add or remove themes from the select, access the Project > Properties menu, there you can add or remove themes for the project.

Log

Defines the log module that will be used in the system. Click here and check out how to create a Log Module.

Menu

Defines the menu application used to create the security item, which contains the applications generated by the module for system management.

This attribute is mandatory for module creation. If the project already has a menu application, they will be listed and can be used in creation. The developer can choose to create a new menu application using the option: **Create application menu**.

Menu orientation

Defines the orientation of the menu that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Menu theme

Determines the initial theme of the menu application that will be created when selecting the *Create application menu* option in the **Menu** attribute.

> If you select an existing menu, this attribute will be disabled

Use SweetAlert

Use SweetAlert to display application messages. When enabled, this option will override the browser’s “confirm” and “alert” settings.

Login

Here you can define the type and number of characters allowed for the username and password fields.

Label Position at Login

Defines the positioning of the fields' Label in relation to the data.

- **Above:** Positions the label above the Above input label at login.
- **Below:** Positions the label below the Down input label at login
- **Watermark:** Positions the label as a watermark.

Login mode

Defines how the user login will be performed.

- **User** - Default form of access, in this option

Cryptography

Use encryption to store the password in the user table. You have the following encryptions available: **MD5, SHA1, SHA256 and SHA512**

Show password characters

Defines whether the reveal password button will be displayed in the password field.

Password field with reveal password button

Use Captcha

It activates the captcha for the login application. The following options are available:

- **No:** Does not display the captcha in the login application.
- **Captcha:** Uses the built-in script library to display the captcha.
- **recaptcha:** Uses Google's recaptcha V2. To configure, see the video below:

  Watch a recaptcha setup video.

Secret Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Site Key

Key generated by Google after creating the recaptcha project.

> This option will be available when using recaptcha. See how to generate these credentials.

Last password change

Defines the time, in days, until the system requests the user to change their password.

> If the value entered is 0, the password will not be reset.

Password strength

Defines the mandatory characters to compose the password for new users.

- **Capital letters**

- **Small letters**
- **Numbers**
- **Special characters**

Example of password strength in the application

Show language selection at login

Enables the display of the language field at login.

Example of language selection at login

Login Field

- **Minimum size**: Defines the minimum number of characters that can be used to define a login.
- **Maximum length**: Stipulates the maximum number of characters for defining a login.

Password field

- **Minimum length**: Minimum number of characters used to define the password.
- **Maximum length**: Defines the maximum number of characters for defining a password.

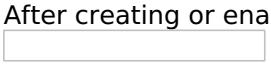
  The maximum values cannot be lower than the minimum values.

Login Template

When enabling the **Use a template attribute on the login screen**, the field for selecting the template that will be used will be available. Templates must be loaded/created in an external library, which in turn must be enabled for use in the project. See how to use an external library

> This option is only available when **Social Network Authentication** is not used.

If you do not have emplates created or enabled for the project, use the button to quickly access the external library settings.

After creating or enabling a template in the external library, if it is not listed, click on the button  to reload the select values.

E-mail Settings

API

This attribute will define whether an API or a customized SMTP will be used for the security module.

SMTP Server

Enter the SMTP server port. This information must be in accordance with the secure connection option.

Door

Defines the connection port according to the SSL used. Use 465 for **SSL**, 587 for **TLS**, or 25 for insecure connection. If you do not enter the port, Scriptcase applies the default port: 25.

Secure Connection:

Use **SSL** or **TSL**, or leave it blank for insecure connection.

SMTP User

Enter user information for SMTP connection. * **EX:** scriptcase@yahoo.com

SMTP Password

Enter the password for the SMTP connection.

SMTP email

Enter the outgoing SMTP email.

Sender's name

Sets the sender name

Email server

Select an email API to use in the security module or define a custom SMTP.

This email will be used in the options:

- All **Recover Password** options
- **New User** when selecting the options: **Requires activation by email**, **Send email to administrator**.

If you do not have an API created for use, check here how to create it.

Customized

Enter all SMTP settings required for operation.

> The options below are only available when the **Custom** option is selected.

- **The SMTP server:** Enter the SMTP server address.
  - **EX:** smtp.mail.yahoo.com
- **SMTP Port:** Enter the SMTP server port. This information must be in accordance with the secure connection option. Use 465 for **SSL**, 587 for **TLS**, or 25 for insecure connection. If you do not enter the port, Scriptcase applies the default port: 25.
- **Secure Connection:** Use **SSL** or **TSL**, or leave it blank for insecure connection.
- **SMTP User:** Enter user information for SMTP connection.
  - **EX:** scriptcase@yahoo.com

- **SMTP Password:** Enter the password for the SMTP connection.
- **SMTP email:** Enter the outgoing SMTP email.

SMTP

> To proceed with the password recovery, it is necessary that has the SMTP server configured for a successful process. Look here how to realize this procedure.

Yahoo

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.
> See also: Ways to securely access Yahoo Mail

SMTP Server Settings Yahoo	
SMTP Server	smtp.mail.yahoo.com
Port	465 or 587
E-mail address	Your full email address (name@domain.com.)
Password	The password for your account.
Requires SSL	Yes
Requires TSL	Yes (if available)
Requires authentication	Yes

Gmail

For more information about these settings Click Here.

> Some email applications uses older, less secure technologies to sign in to your email account, and Google will block these requests by default. To solve this, sign in to your Google account and enable access through less secure apps through this link.

SMTP Server Settings Gmail	
SMTP Server	smtp.gmail.com
E-mail address	Your full email address (Ex. example@gmail.com)
Password	The password for your account.
Port (TLS)	587
Port (SSL)	465
TLS / SSL Required	Yes

Outlook / Hotmail

For more information about these settings Click Here

SMTP Server Settings Outlook.com	
SMTP Server	smtp-mail.outlook.com
E-mail address	Your full email address (Ex. myname@outlook.com, not an alias)
Password	The password for your account.
Port	587 (you can use port 25 as an alternative)
TLS / SSL Required	Yes

New users

Settings for creating new users in the security system.

New Users

Allows new users to register

This option defines the availability of user registration directly in the login system. If you do not check this option, only users with administrative access can add new users.

Requires email activation

This option defines whether the new user must validate their registration by email to activate the account (Configure SMTP email to use this option)

Send email to administrator

This option defines whether the system administrator receives an email whenever a new user is registered. (Configure SMTP email to use this option)

Default Group

Defines the default group that the user will be inserted into when created.

> This feature is only available for the group security module or LDAP/Applications

Recover password

Send email with password

The system sends the user’s password via email only if SMTP has been configured correctly, and the password is not using encryption.

Reset password and send new one by email

The system automatically resets the password and sends it to the user via email (only if SMTP has been configured correctly).

Send email with a link to generate a new password

The system sends the user a link to access an application and reset the password.

> The email sending options, for example, *Send email with password* or *Requires activation by email* requires the correct configuration of the administrator’s email in the **Email Settings** tab.

Two Factor Authentication

Two-factor authentication (2FA)

Determines whether or not two-factor authentication is used in the system. Currently, the security module supports three ways of configuring resources:

- **Authentication**: This option will send the code to your Google authentication app.
- **SMS**: This option will send the authentication code via SMS.
- **Email**: This option will send the authentication code to the configured email.

Operation mode

In this attribute, the developer can define whether users will be required to use authentication to access the system or whether they may not use the resource.

- **All** - All users are required to configure
- **Individual** - Each user has the option of using authentication or not.

API

Lists the available APIs according to the type of authentication selected in the *Two-factor authentication (2FA)* attribute.

If you do not have any API created for the project, use the button to quickly access the API settings.

After configuring the API, click the to reload the select values.

> If you do not have an API created for use, check here how to create it.

Code expiration time

Set the time in **seconds** that the 2FA authentication token will expire.

,

Ldap

In the LDAP configurations, it is necessary to enter the access data for the previously configured server.

Server

Enter the LDAP server IP here.

DN

Enter the input attributes.

Door

Enter the server access door here. **The default port is 389.**

Suffix

Enter the user suffix.

Social Network

The Scriptcase allows user authentication through social networks such as **X** (formerly Twitter), **Facebook**, and **Google**. This feature makes application access easier by eliminating the need to create new credentials, making login more practical and secure.

To enable login with a social network, it is necessary to configure an application on the corresponding platform to obtain the access credentials.

> When using social network authentication, the **Login Template** option will not be available.

Facebook

Authentication via **Facebook** allows users to log in to the system using their social network accounts. To enable this feature, it is necessary to register an application on **Facebook for Developers**. See how to Generate Facebook credentials.

To use **Facebook**, check the **checkbox** to enable the **App ID** and **Secret** fields and enter the previously generated data.

App ID

Enter the **App ID** available in the settings of the application created in **Facebook Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **Facebook Developers**.

X (formerly Twitter)

Allows user authentication via **X** (formerly Twitter), using the platform's API. To configure this feature, it is necessary to create an application in the **X Developer Portal** to obtain the authentication credentials. See how to Generate X credentials.

To use login with **X**, check the **checkbox** to enable the **Key** and **Secret** fields and enter the previously generated data.

Key

Enter the **Key** available in the settings of the application created in **X Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **X Developers**.

Google

Login via **Google** allows users to authenticate using their platform accounts quickly and securely. To enable this feature in Scriptcase, it is necessary to configure an application in **Google Cloud Console**. See how to Generate Google credentials.

To use login with **Google**, check the **checkbox** to enable the **Client ID** and **Secret** fields and enter the previously generated data.

Client ID

Enter the **Client ID** available in the settings of the application created in **Google Developers**.

Secret

Enter the **Secret** available in the settings of the application created in **Google Developers**.

Logged Users

This setting defines the system behavior for login protection.

This option is available if you checked **Protect logged in users** during the connection step.

Show logged in users

If you select this option, Scriptcase will also create a Grid Application with the Security Module to display a report with all users currently connected to the system.

Brute Force attack protection

Allows/disables blocking users after a few failed access attempts.

Brute Force block time (in minutes)

Time, in minutes, that the user will remain inaccessible after several failed access attempts. (Only available when Protection against Ruth Force Attacks is activated)

Number of attempts before blocking

Number of failed access attempts until protection is enabled. (Available only when protection is enabled for brute force attacks)

Insert Data

This option inserts the first user into the security tables. This user has administrator privileges to manage the system and add new users.

Login

This option defines the system administrator user. By default, the user defined by Scriptcase is **admin**.

Password

This option sets the system administrator password. By default, the user defined by Scriptcase is **admin**.

> To increase the security of your project, we recommend changing the default username and password “admin” at this stage or after the first access.

Name

This option defines the name of the administrator user on the system.

Email

This option defines the email address of the system administrator user. This email can be used for some new user registration settings, such as notification to the administrator.

> For the email to be sent successfully to the administrator, the SMTP parameters must have been configured or an email API must have been selected in the previous step in Configurations.

Add Applications

This option adds the applications already created in the project to the Security Module applications table.

Include All

All project applications will be added to the application table.

Select Applications

Allows the developer to select specific applications to include in the application table. When clicking Select, all existing applications in the project will be listed.

Application selection screen

On this screen you can search for applications by name or filter by their types.

Once selected, the applications will be listed below.

List of selected applications

When unchecking this option, existing applications in the project when creating the module will need to be inserted through application synchronization, after executing the security module.

Save Profile

This feature will always be displayed when completing the security module configuration. The saved profiles will be displayed on the initial module creation screen, in the **Create / Choose Profile** step.

Viewing Saved Profiles

Ao clicar em **Usar**, o processo de criação do módulo de segurança se dará da mesma forma, porém, com as opções de configuração preenchidas com os valores salvos no perfil. When clicking on **Use**, the security module creation process will take place in the same way, however, with the configuration options filled in with the values saved in the profile.

Save Profile

When enabled, the settings made in the created module will be saved in the profile.

Name

When enabling profile saving, the name becomes a mandatory field to identify the created profile.

Level

Defines the viewing scope in which the profile will be saved.

- **Public:** Defines that the security profile will be available in any Scriptcase project.
- **Project:** Defines that the security profile will only be available in the project in which it was created.
- **User:** Defines that the security profile will only be available to the Scriptcase user who created it.

Building Security

Finally, a summary of the creation of the security module, listing the changes made by Scriptcase to your project.

- **Open Project:** You will be redirected to the home page of the project being modified.
- **Generate source code:** Generates the source code of all applications created by the security module.
- **Download SQL:** Makes the table structures of the created module available to the developer

  After creating the security module, it will be necessary to generate the source code of all applications that were marked by the security module usage flag. Except Login and Menu applications.

Log Module

Using the log module, you can create multiple log schema, each project application can be linked to a different log schema, which will be configured individually. Each project can have one or more log schemes applied to its applications.

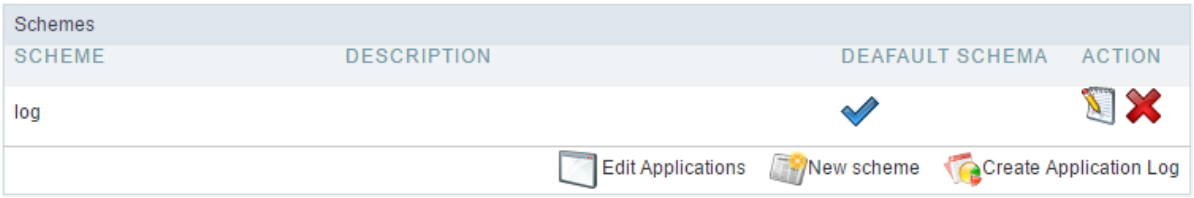
Schema list

Access the menu Modules > Log and chick **Create / Edit Log Module**.

If you are creating your first log schema project, proceed direct to step three, for the schema Creation/editing.

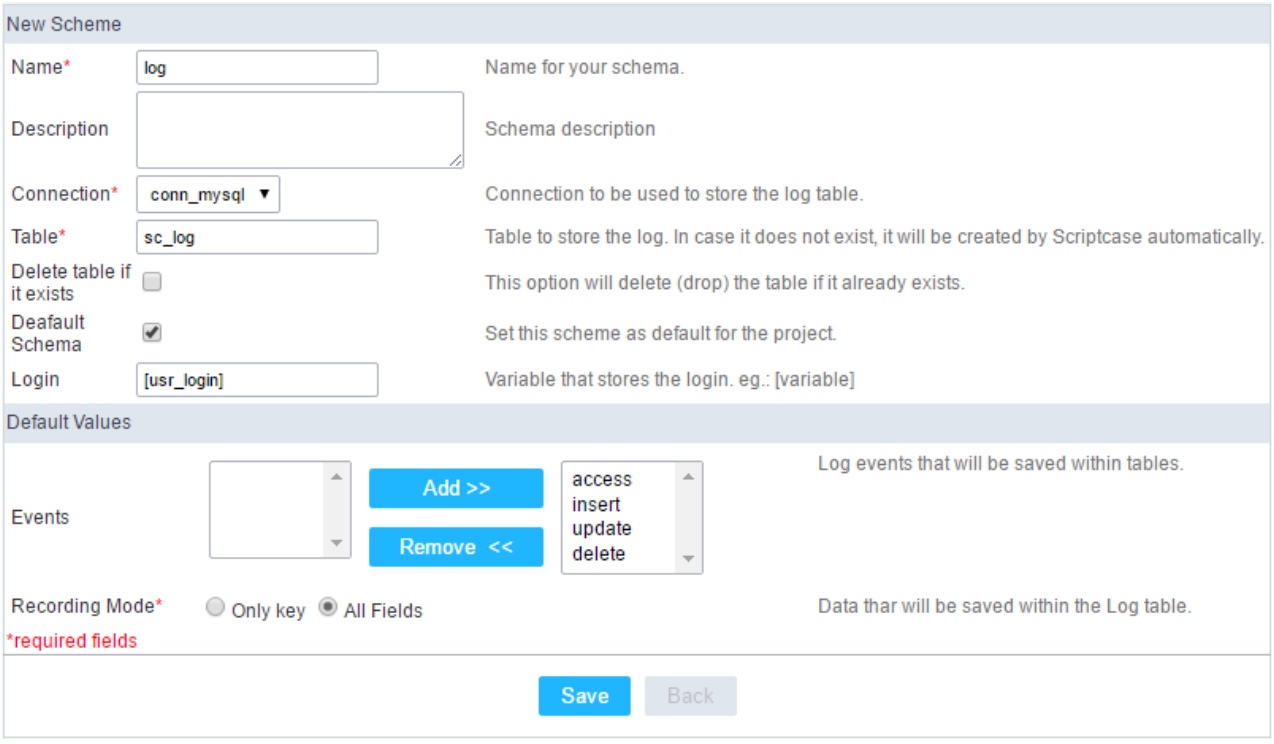
It lists all log schemes previously created. On this screen, we can manage the schema created.

- **Edit Applications** - Edits the relation between the applications and the schemes.
- **New scheme** - Creates a new Log scheme in the project.
- **Create Application Log** - Creates a report using the log table, that is also used by the project's default scheme.
- **Edit Scheme Icon** - Edits the Log scheme.
- **Delete Scheme Icon** - Deletes the Log scheme.



Schema Creation/editing

Access the menu Modules > Log and check **Create / Edit Log Module**.



New Schema

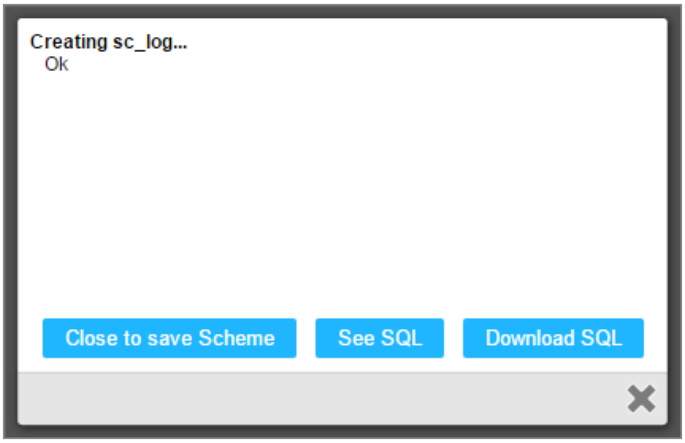
- **Name** - Log schema name that will be created. It is a mandatory field.
- **Description** - Short description for the new schema.

- **Connection** - Connection on where the log table will be created. It is a mandatory field.
- **Table** - Table name on where the log information will be stored. It is a mandatory field.
- **Existing Table** - This option deletes the table that has the same name in the database and create a new one. If you want to use the existing table to store the log data just leave this option unchecked.
- **Default Schema** - It sets the schema that is being created as the default one for project.
- **Login** - Global variable used to store the logged user information into the system. This variable must be the same used in the Security Module. By default, Scriptcase sets this variable on both modules as `[usr_login]` .

Default Values

- **Events** - Actions that will be stored in the log table.
- **Storage mode** - User data that will be stored in the log table.

A confirmation screen will appear when you save, just click **Close** to leave and save the schema.



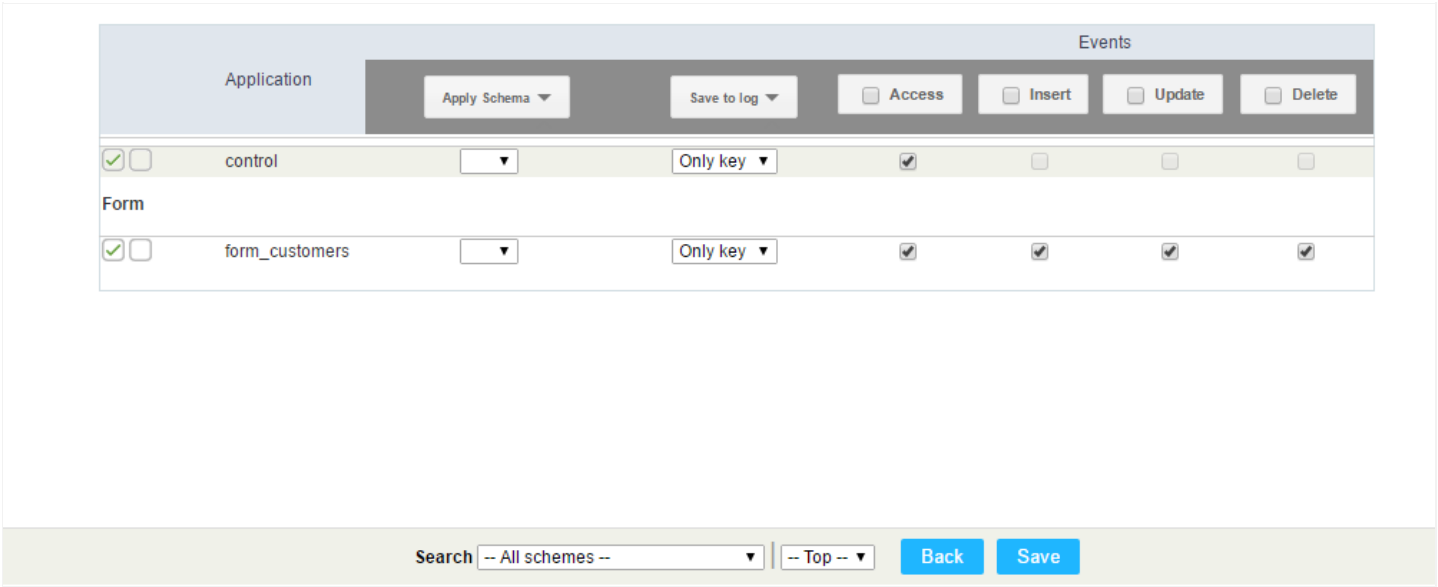
- **Close to save the schema** - Saves the schema you created, and redirects to the applications screen.
- **See SQL** - Display the SQL used by Scriptcase to create the table 'sc_log' that will be used to store the log events.
- **Download SQL** - This option will download the SQL used by Scriptcase to create the table 'sc_log'.
- **Close Icon** - This option cancels the schema creation and redirects to a new schema creation.

After the confirmation, you will be redirected to link the log schema to the project applications.

Link Applications

In this step, you must define which log schema will be used in each project application. The list of applications can be filtered by type or by log schema used.

At the top, you can apply actions to all rows or you can select only one action for each row individually.

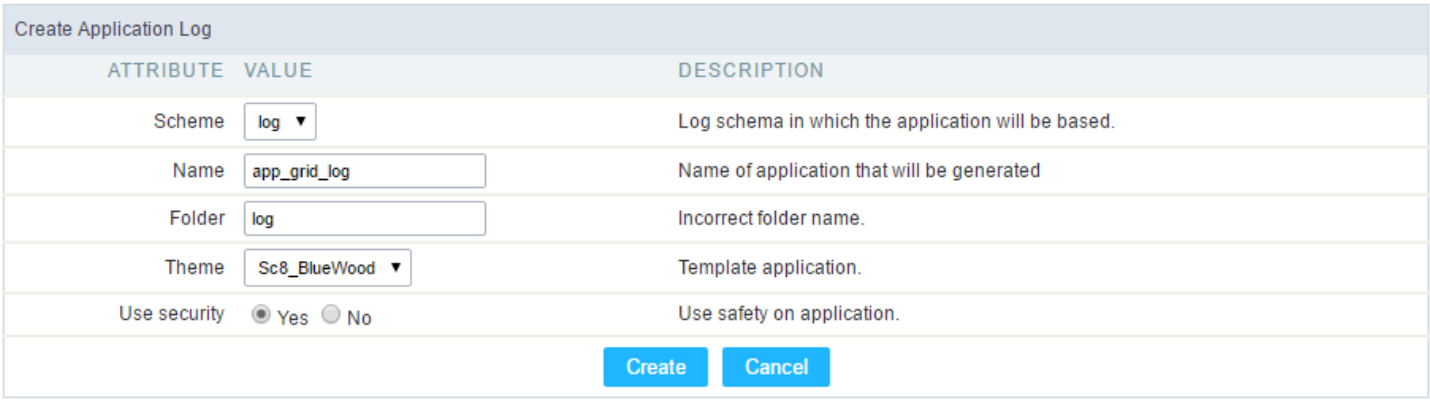


- **Apply schema** - Select the schema that will be used for the applications. (when you select a schema, the other information is filled according to what was previously configured).

- **Store for the Log** - User data that will be stored in the Log table.
- **Access** - This event stores the information every time the application is accessed.
- **Insert** - This event stores the information when inserting a record.
- **Update** - This event stores the information when updating a record.
- **Delete** - This event stores the information when deleting a record.

Log Report

In this report, you can see all stored information according to the schema selected during the report creation.



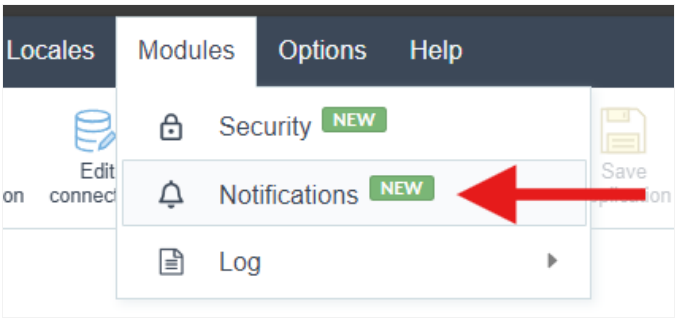
- **Schema** - Log schema that the application will be based on
- **Name** - Application name
- **Folder** - Folder name where the application will be stored
- **Theme** - Theme used for the generated application
- **Security** - This option attaches a security profile to the log schema.

Notification Module Overview

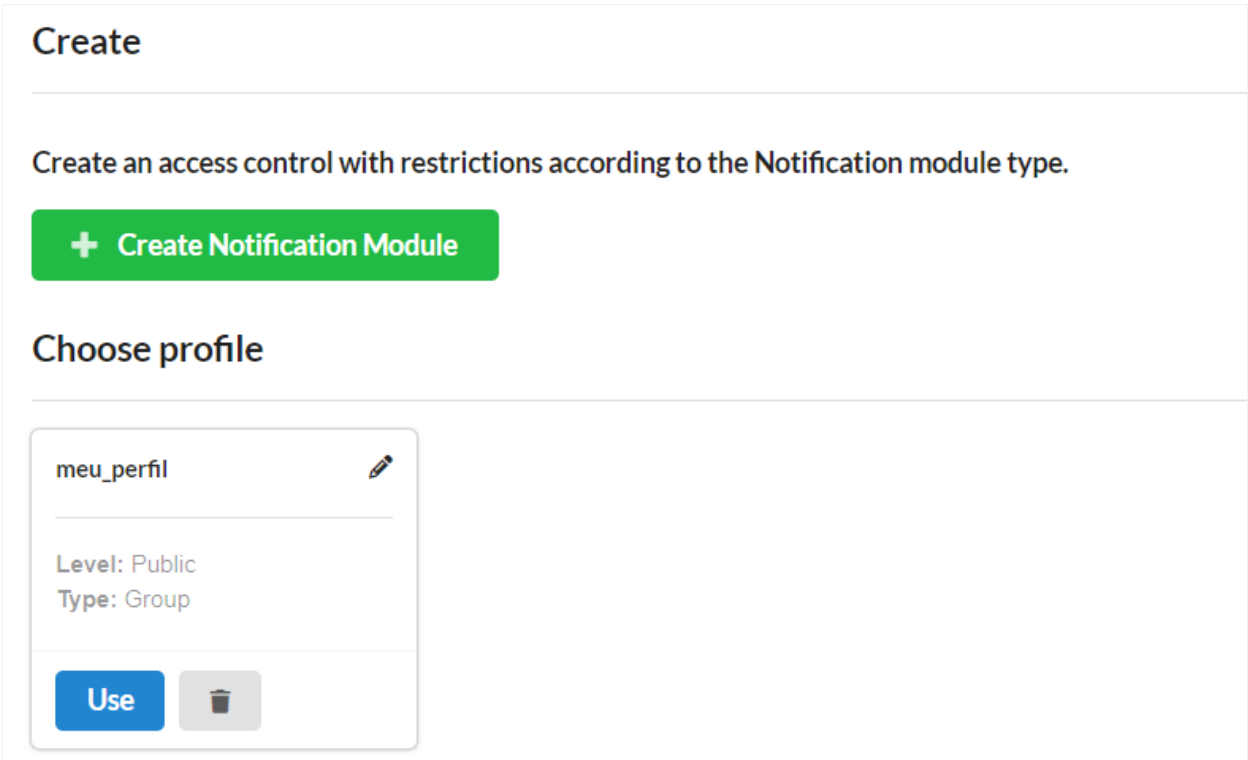
The Notification Module automatically generates a set of applications used in the management of system notifications. This module allows you to create access control with restrictions according to the type of Notification module.

Create / Choose profile

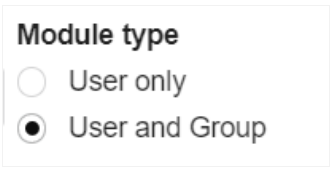
Access the menu and click on: **Modules > Notifications.**



On this screen, you can create a module from scratch or choose a previously saved profile.

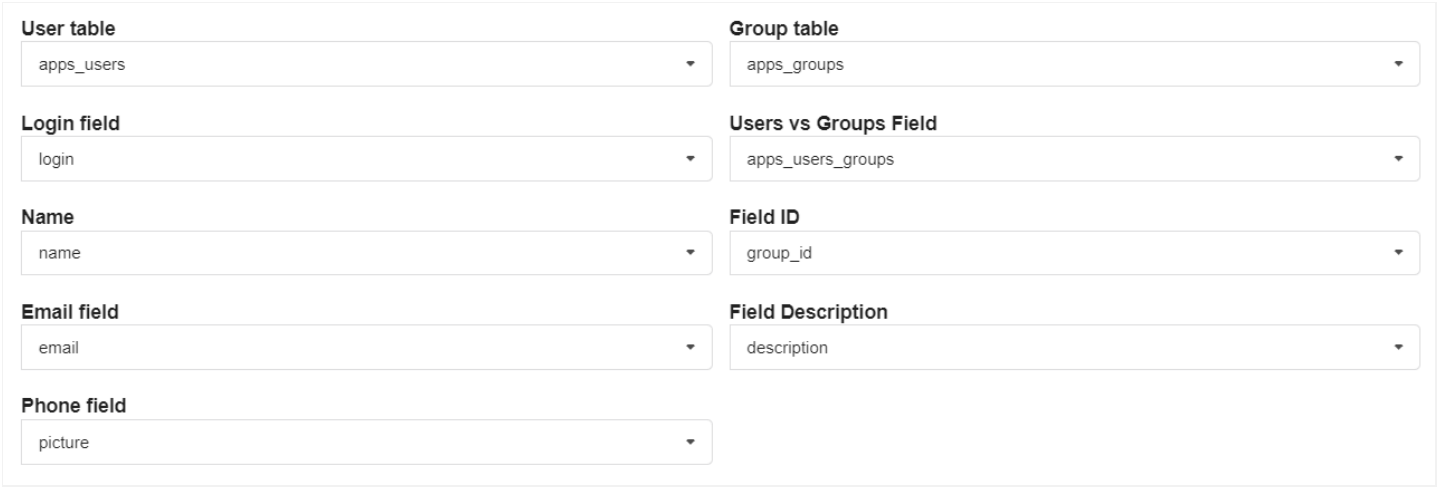


Module Type



User Only: Creates a notification module based on the user

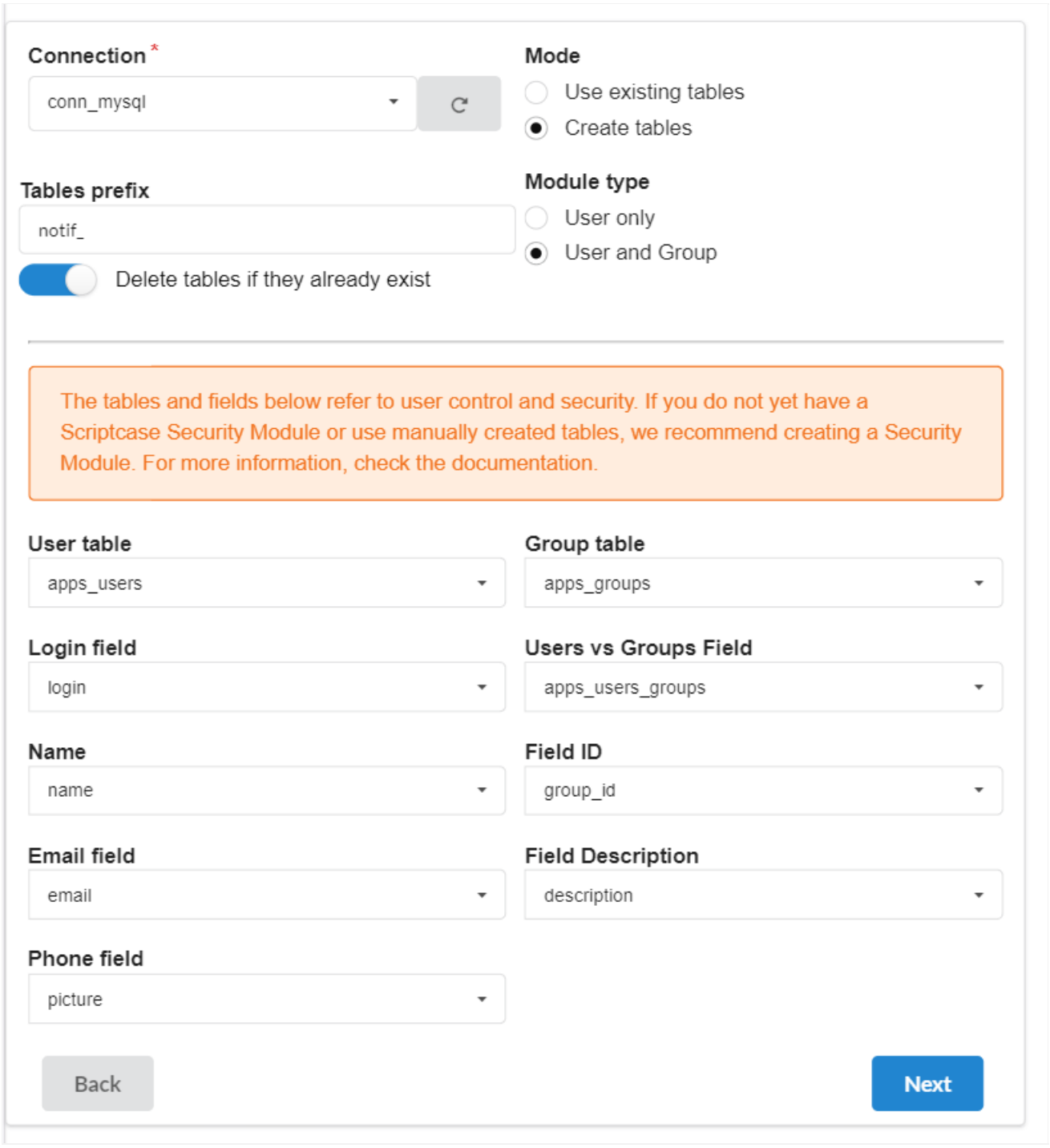
User and Group: Creates the notification module based on users and groups.



Connection Settigs

Note: It is recommended to create the security module beforehand.

Here you must choose which connection the module will be created in and define the other settings.



Connection

Choose the connection in which the module will be created.

Mode

If you already have notification module tables in your database, you can choose to use them. If you do not have them yet, you can choose the **'Create tables'** option.

Click here to learn how to use existing tables.

Table prefix

Choose the prefix for the tables.

Module type

Choose the option according to the security module created: User or User and Group

Delete tables if they already exists

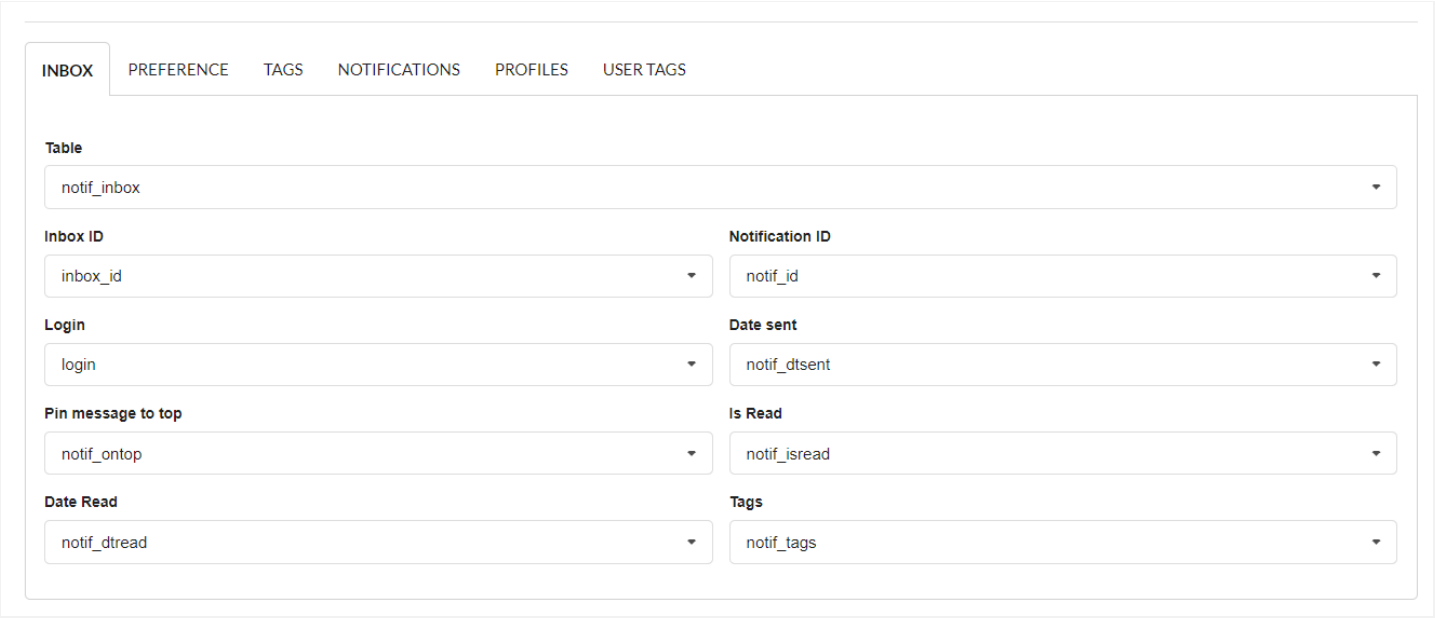
If enabled, the module will remove the tables if they already exists.

The fields that reference the **users, groups, login, users x groups, name, capo id, email field, description field and phone field** tables will automatically be according to the module type.

Associate tables

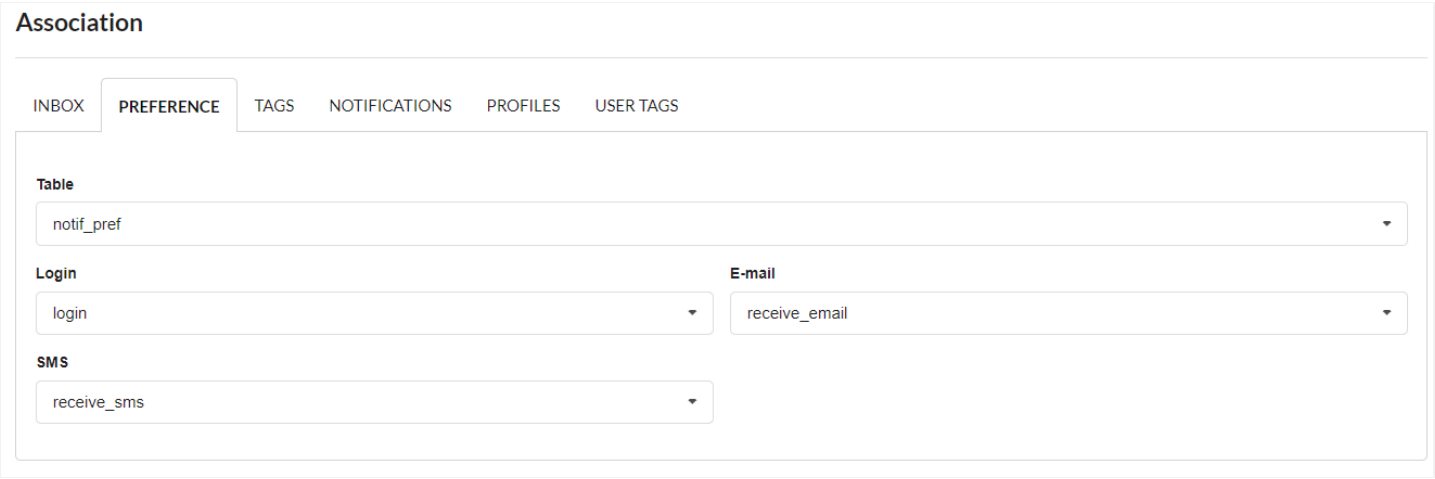
When choosing to use existing tables for the notification module, you must associate them manually.

See below the structure of the notification module tables



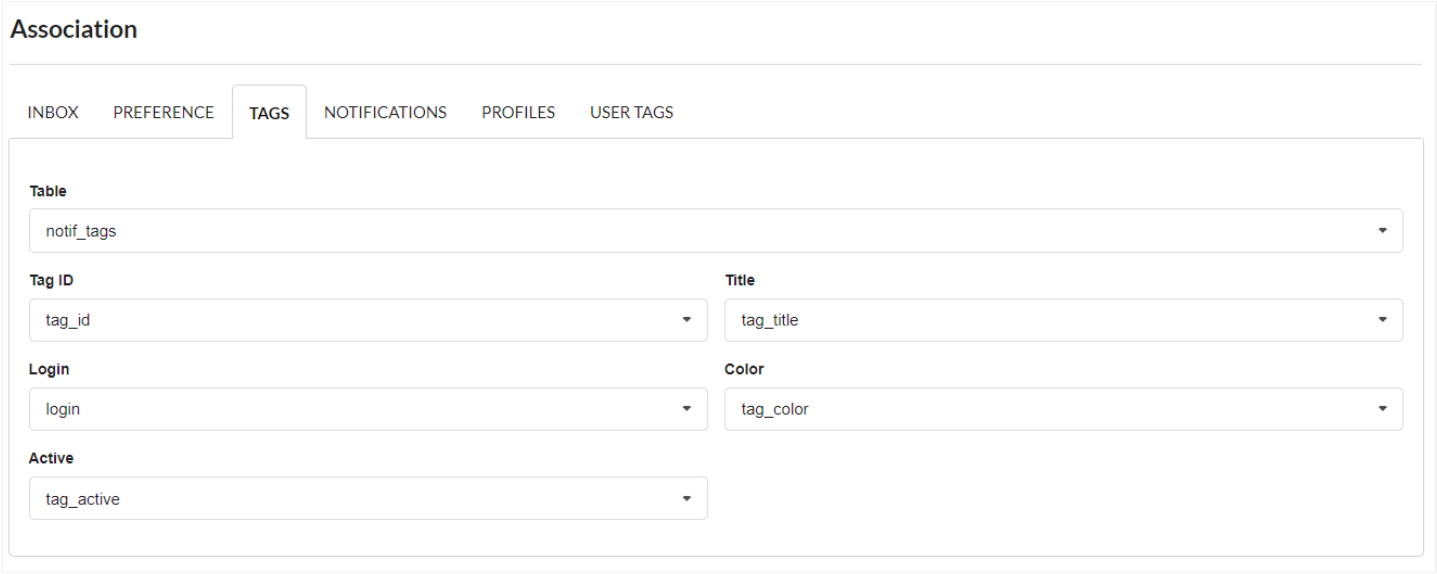
notif_inbox

```
CREATE TABLE `notif_inbox` (
  `inbox_id` int NOT NULL AUTO_INCREMENT,
  `notif_id` int NOT NULL,
  `login` varchar(255) NOT NULL,
  `notif_dtsent` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `notif_ontop` int NOT NULL DEFAULT '0',
  `notif_isread` int NOT NULL DEFAULT '0',
  `notif_dtread` datetime DEFAULT NULL,
  `notif_tags` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`inbox_id`)
)
```



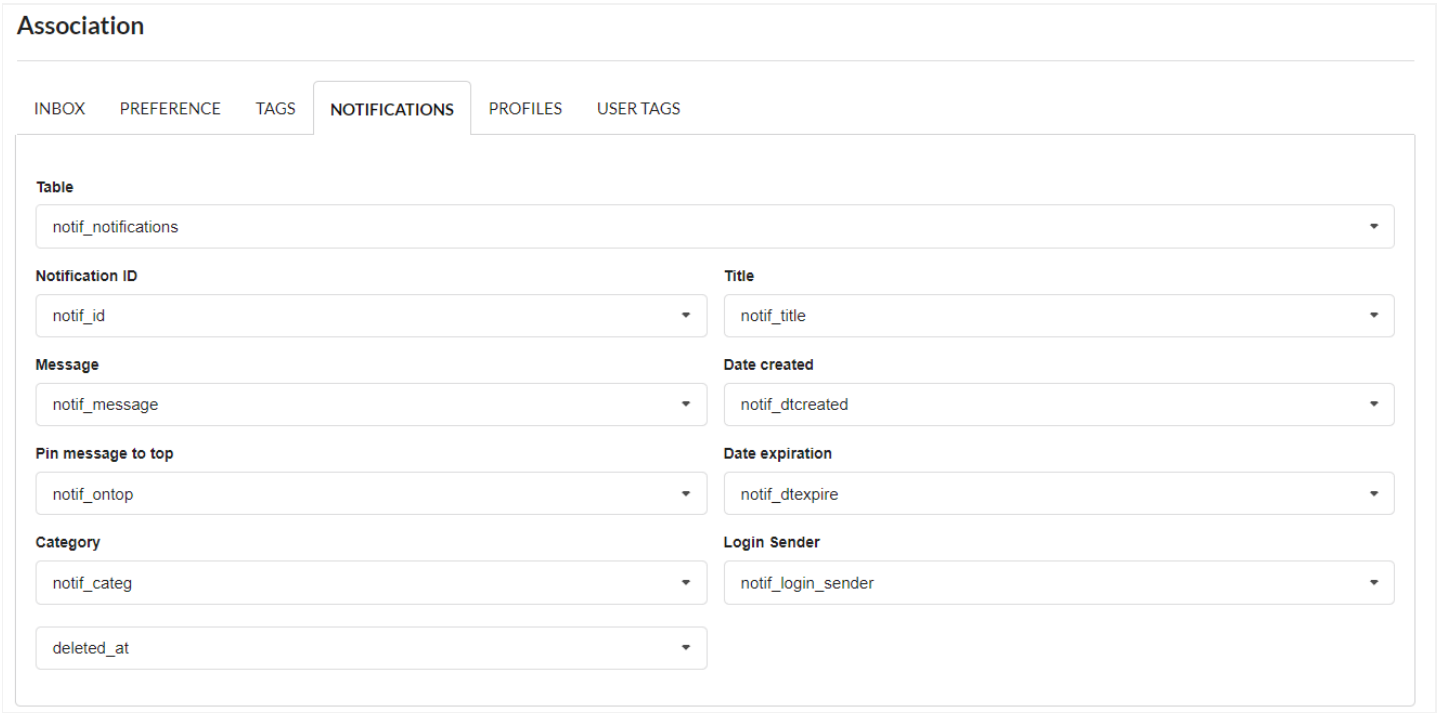
notif_pref

```
CREATE TABLE `notif_pref` (
  `login` varchar(255) NOT NULL,
  `receive_email` int NOT NULL DEFAULT '1',
  `receive_sms` int NOT NULL DEFAULT '0',
  PRIMARY KEY (`login`)
)
```



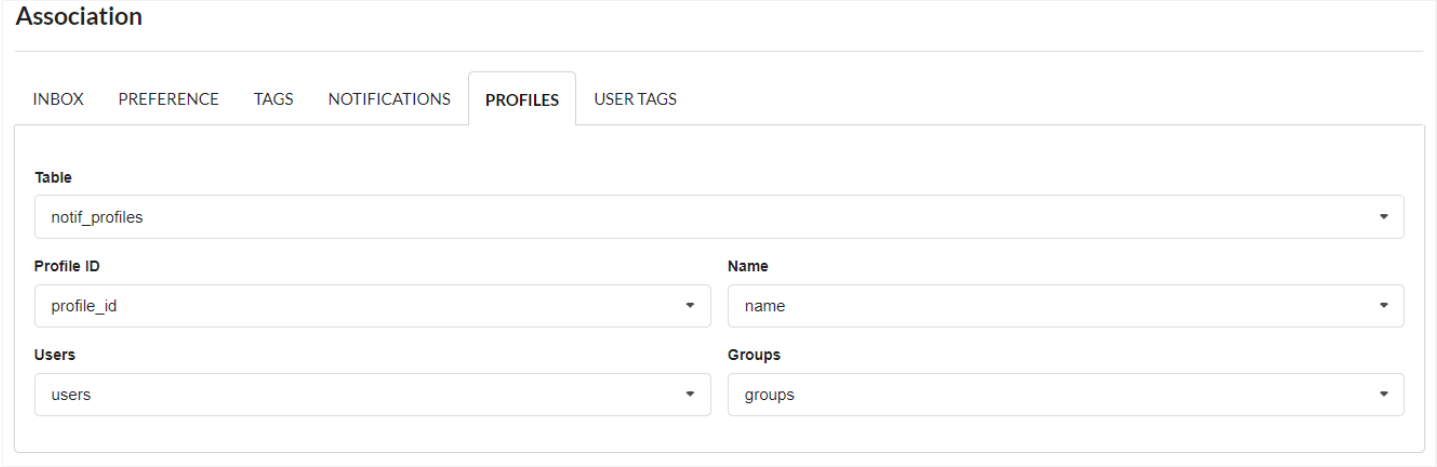
notif_tags

```
CREATE TABLE `notif_tags` (
  `tag_id` int NOT NULL AUTO_INCREMENT,
  `tag_title` varchar(50) NOT NULL,
  `login` varchar(255) NOT NULL,
  `tag_color` varchar(100) DEFAULT NULL,
  `tag_active` int NOT NULL DEFAULT '1',
  PRIMARY KEY (`tag_id`)
)
```



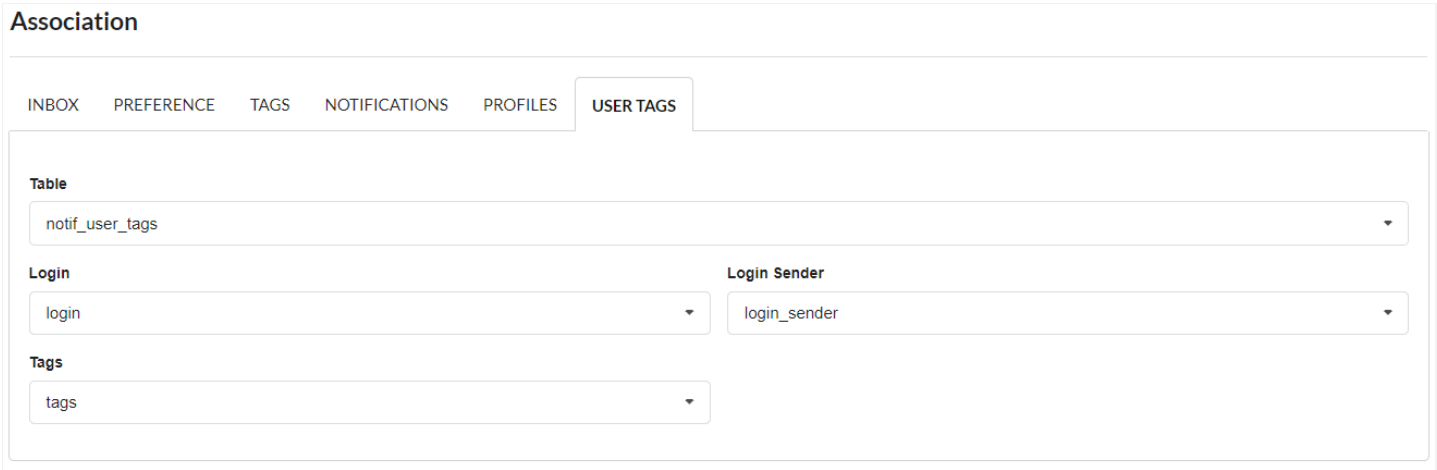
notif_notifications

```
CREATE TABLE `notif_notifications` (
  `notif_id` int NOT NULL AUTO_INCREMENT,
  `notif_title` varchar(255) NOT NULL,
  `notif_message` text NOT NULL,
  `notif_dtcreated` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `notif_ontop` int NOT NULL DEFAULT '0',
  `notif_dtexpire` datetime DEFAULT NULL,
  `notif_categ` varchar(60) DEFAULT NULL,
  `notif_login_sender` varchar(255) NOT NULL,
  `deleted_at` datetime DEFAULT NULL,
  PRIMARY KEY (`notif_id`)
)
```



notif_profiles

```
CREATE TABLE `notif_profiles` (
  `profile_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(255) DEFAULT NULL,
  `users` text,
  `groups` text,
  PRIMARY KEY (`profile_id`)
)
```



notif_user_tags

```
CREATE TABLE `notif_user_tags` (
  `login` varchar(255) NOT NULL,
  `login_sender` varchar(255) NOT NULL,
  `tags` varchar(255) DEFAULT NULL
)
```

General

In this step, General Settings, Email, SMS and Insert Applications will be available

Below you will see each configuration option in detail.

Application prefix

allows you to define the prefix that will be given to the applications. By default, the prefix will be**notif_**

Folder

allows you to define the name of the folder that will be responsible for storing the applications related to the notification module.

Theme

Allows you to choose the theme of the applications of the notification module

Log

Will display the log module if you have any module created.

Menu

Allows you to choose between the options:

- – Create menu application – : When choosing this option, a menu will be created by the module. This option will enable the Menu theme and menu orientation options.
- Menu application: If you have a menu application created separately or by the security module, you will be able to define it.

Menu theme

If you want to create a menu application, this option will be enabled, allowing you to choose which theme will be defined for the menu.

Menu orientation

If you want to create a menu application, this option will be enabled, allowing you to choose which menu orientation.

Email Settings

In this tab, you define the email sending settings.

API

Allows you to choose between 3 options:

- **Disabled:** Ignores the use of email API
- **Use the same as the security module:** Allows you to use the same email settings as the security module
- **Customized:** Enables the other fields so that you can manually define the email credentials.

SMTP server

Defines the email SMTP server.

SMTP port

Defines the port to be used.

Secure connection

Defines whether the connection will be secure or not: SSL or TLS

SMTP user

Defines the SMTP user

SMTP password

Defines the password

SMTP E-mail

Defines the email to be used when sending email messages

SMTP sender name

Defines the name that will be displayed to the sender when sending emails.

If you do not have an API created for use, check here how to create it.

SMS Settings

In this tab, the SMS sending settings are defined.

API

If you have already created an sms api, you can define it in this option, otherwise, the disabled option will be the only one displayed.

Number of characters to limit SMS

This option allows you to define the number of characters in the SMS message

If you do not have an API created for use, check here how to create it.

Insert Applications

In this tab, you can add applications to the notification module and the security module’s application tables.

Include all

With this option enabled, all applications contained in the module will be included in the application table.

Select table

Allows you to select the notification module and security module tables that will be related.

Save Profile

Using this option, you save all the settings made during the creation of the Notification Module. With the profile, the settings of this module can be used later for other projects.

Save Profile

Allows you to save a profile with all the current settings of the notification module.

Name

Name that will be given to the profile with these settings. It will help to identify the profile later.

Level

This option allows you to select the viewing scope in which the profile will be saved. There are three options:

- **Public:** Defines that the notification profile will be available in any Scriptcase project.
- **Project:** Defines that the notification profile will only be available in the project in which it was created.
- **User:** Defines that the notification profile will only be available to the Scriptcase user who created it.

Options Overview

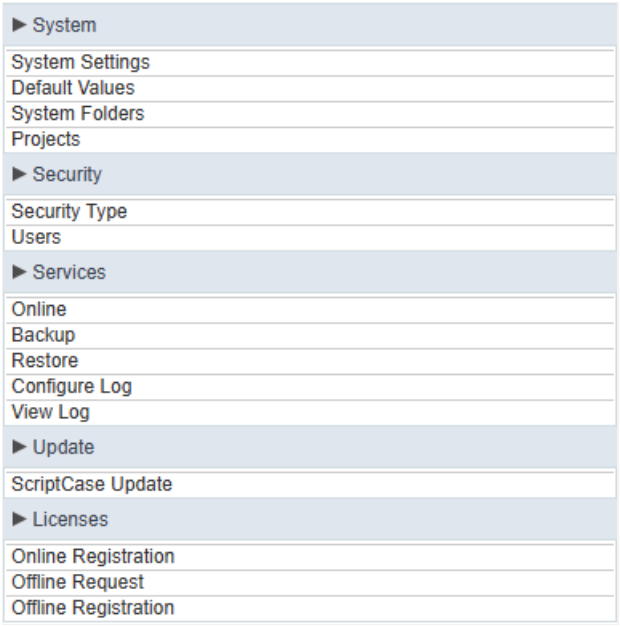
Gathers the main configurations of ScriptCase. For the users with administrator privileges, it's possible to register the ScriptCase, manage the system users, do backups and restore projects and other configurations.

For the other users, without administrator privileges, its possible to define the default language for ScriptCase, customize Scriptcase's toolbar and other options that when done, will be available for the users that did them, for those who won't change these configurations, they'll use the default configurations from ScriptCase.

Admin Settings Overview

The **Settings** menu brings together the tool's management settings and is accessible only to users who have administrator privileges in Scriptcase.

The **admin** user, created by default when installing the tool, has administrator privileges. As this is a standard user, it is recommended to change the password, as well as set the SMTP so that the **Forgot your password?** option works correctly.



Check the table below with the link to the specific documents for each option.

System	Security	Services	Update	Licenses
System Settings	Security Type	Online	Scriptcase Update	Online Registration
Default Values	Users	Backup		Offline Request
System Folders		Restore		Offline Registration
Projects		Configure Log		
		View Log		

System Settings

Here you can define the settings about the Scriptcase development environment.

Quantity of compilations in parallel

Defines multiple threads to generate applications. The default value is 5, so it can compile five applications at the same time.

Display the compile-time applications individually

Allows displaying the compilation time for each application individually.

Number of automatic application copies

Scriptcase stores versions of each application automatically. Here, you can set the limit of backup copies it keeps. The older one is overwritten.

Send an alert if the last backup is older than(days)

It allow you to set a number of days to alert when the last Scriptcase backup is obsolete.

Show backup alert for

Defines the user’s group to receive the backup alert.

Check for ScriptCase update versions after login

Defines the period to check if there is a new Scriptcase update released.

Performing Scriptcase’s database verification

Defines the period to check the backup of ScriptCase’s database (Not the projects).

ScriptCase Session Timeout (min)

The time limit for executing PHP scripts during application development. For example, to compile or save applications. (in seconds).

Time to expire the login cookie (days)

The quantity of days Scriptcase keeps the cookies of login.

Check IP when log on via cookie

Scriptcase uses the cookies to validate the IP of the logged user.

Check Browser when log on via cookie

Scriptcase uses the cookies to validate the browser of the logged user.

Check Machine when log on via cookie

Scriptcase uses the cookies to validate the computer configuration of the logged user.

Show Generation Status Message

Allows you to define the action messages of ScriptCase, when it is compiling or running your applications, in the options:

- **Main**: Displays a subgroup of the messages.
- **None**: Does not display the messages.
- **All**: Enables the display of all the messages.

Use cache for connection metadata

Seeks better performance without having to read the database dictionary every time applications are created. This information is added in files, so that there is a more agile creation in the recovery of existing metadata in the databases.

Save metadata cache on disk

Allows files to be created in a directory, from the project name, containing a lifetime to be recreated.

Cache lifetime in days

Sets the cache lifetime for files to be recreated. In addition to this option, files will also be created in the following situations:

- When editing connections, if there is any change in the database, the cache is recreated.
- The “reload” icon next to the tables.
- The SQL Builder “reload” button.

SMTP Server

You should inform SMTP server address for the provider.

SMTP Port

Port used by the mail server. Use port 465 for security with SSL, port 587 for security with TLS, or port 25 as port without security. By omitting the value, Scriptcase defaults to 25.

SSL

Defines if it uses SSL security or not.

SMTP User

Inform the SMTP server user.

SMTP Password

Inform the SMTP server password.

SMTP E-Mail

Inform the sender e-mail. Who is sending the emails.

PROXY Server

Inform the IP of your Proxy Server.

PROXY Gateway

Inform the Gateway of your Proxy Server.

PROXY User

Inform the proxy user.

PROXY Password

Enter the proxy user password.

Use session in database

It allows us to store the session data into the database.

Show an option on the deployment wizard to store the PHP session variables in a Database

It allows displaying in deployment, an option to store the session into the database.

Pass the PHP Session ID in the URL

It allows passing the PHP Session ID in the URL.

Foreign key improvement

When it comes to the form, this enables an improvement in the definition of lookups in a field that is a foreign key.

Example of a lookup created when the flag is enabled:

```
SELECT orderid, customerid FROM orders ORDER BY customerid
```

Example of a lookup created when the flag is disabled:

```
SELECT orderid, orderid FROM orders ORDER BY customerid
```

When it comes to grid applications, this enabled flag creates automatic lookups for fields. Regarding the data dictionary, with this option enabled, the data dictionary will now allow the foreign key to be defined when editing fields, more precisely in the additional properties of the fields.

Timezone

It sets the PHP Timezone used in ScriptCase.

Default Values

With this configuration, it is possible to standardize the projects' creation of the applications, increasing the applications' productivity in development. If you wish to change the settings for a specific project, access the menu **Project > Default Values**.

Common Settings

Logo

The Favicon to represent the project in the browser. It uses the same Favicon for all applications of the project.

Records per page

Amount of records per page.

Theme

The default selected Theme for the projects.

Show Summary

Displays the number of records on the page and the total of the records in the application. This option is able in the application toolbar.

Use template in the HTML Editor

Selecting Yes, it uses the new template of the HTML Editor field. Set the themes through the menu **Layout > HTML Editor Templates**.

HTML Editor template

Defines the template for editing the field HTML Editor.

Initial sort fields

Text

Set the initial sorting for the text fields.

Date

Set the initial sorting for the date fields.

Number

Set the initial sorting for the number fields.

Currency

Set the initial sorting for the currency fields.

Grid

Table Width

It defines the length of the application table. This option is available to select **Pixel** or **Percent** in the table width unit.

Table Width Unit

The unit used to define the application's width. Automatic (Automatic Width according to the fields' size);

Pixel (Set the width in pixels. Example: 800 pixels); Percent (Set the width in percentage. Example: 80%)

Tab a Group By

The left margin of the Group By.

Separates the Group

The margin between two Group By.

Lines per page

List of options for the user to select the quantity of the records per page. If you need an option to list all the records, you should ad “all” to the options. Ex.: 10,20,30, all

Form

Table Width

It defines the length of the application table. This option is available to select **Pixel** or **Percent** in the table width unit.

Table Width Unit

The unit used to define the application’s width. Automatic (Automatic Width according to the fields’ size); Pixel (Set the width in pixels. Example: 800 pixels); Percent (Set the width in percentage. Example: 80%)

Tab

Tab Alignment

The alignment of the tabs in the application.

Menu

Horizontal Alignment of the Menu

Menu alignment.

Horizontal Alignment of the Items

Menu item alignment

Template

It defines the default templates of the Header and Footer.

Toolbar Buttons

It defines the default buttons in the toolbar for the new applications.

System Folders

Here you can set the directories path to stores the Scriptcase data information. Scriptcase uses the default paths when using the installer or the **Typical** manual installation method. You don’t need to make changes. We recommend contacting the support if you wish to change them.

Root Directory for the Web Server files

Set the Root of the Web Server. On **IIS servers**, you should use the path: “inetpub/wwwroot”.

Application Path

Set the folder that stores the generated applications source code.

Production Path

Set the folder of the Scriptcase “prod” files. It keeps all the libraries needed to execute the generated applications.

Documents Directory

Set the folder to store document files used in applications.

Images Directory

Set the folder to store image files when using the upload field (Image File Name) of applications.

Temporary Files Directory

Set the folder of temporary files used to manipulate the images and the exporting data of the applications.

Projects

List of Scriptcase existing projects. You can view all project details, such as users who have access, the current version of projects, number of applications, among other information.

Project

Project name.

Description

Project description.

Creation

Project creation date.

Applications

The number of applications from each project version. 1:33(Versions count: number of applications)

Action

Project details.

By clicking on the magnifying glass next to the number of applications, you can see more details of the project. In addition to some information previously views on the list of projects.

Connections

Lists of connections of the project.

User

List the users that can access the project. Default permissions, like create applications.

Administrators

User/Groups that are administrators of the project. It’s possible to create and edit connections in the properties of the project, with this permission.

> You may need to set the permission of each user individually as the group permission also.

Security type

Select the way how Scriptcase users can access the interface.

Security Type

The method of data storage. We can select the following methods:

Scriptcase

It stores the access data into the ScriptCase's database.

LDAP

It stores the access data into the LDAP Server and Scriptcase goes there to validate the access.

Security Level

Select the type of security. this option is independent of the storage method.

User

It defines the permissions for each user. You must configure them individually.

Group

It defines the permissions for the User Groups. Here, we must associate the users to one or more groups and set the permissions for each group.

User/Groups

It defines the permissions by the groups and by users. The grant by the user overwrites the group grants. It allows having users existing in the same groups but with different permissions.

> The security level option is the same independently of the Security Type. (Scriptcase or LDAP)

Scriptcase

Managing user groups. You will see a list of the existing groups and some information about them.

> This option will only be available on the left side menu when we select the Groups option in the **Security Type**

Name

Name of the groups.

Description

Group Description.

Users

The users from this group.

Projects

Set the projects that the users of this group can access.

Actions

Users

Let you select the users for this group.

Edit

Let you edit the group information.

Delete

Delete the group.

You will see more details about the existing options.

New Group

Create new groups. We can define the privileges of the groups.

Name

The name of the group.

Description

The group description.

Admin Privilege

It defines if the users of the group have Administrator Privileges.

Privileges

Project

Let you set which actions the group users can do when accessing a project.

Application

It defines the permission to create applications.

Connections

It defines if the users can manage the database of the project.

Projects

Set the projects that the group users can access.

Admin projects

It defines if the group have Administrator grant for the project.

Users

After creating the group, we can define which users are part of the group.

LDAP

This page let you sync with LDAP. We recommend using the LDAP Administrator user, this way, it lists all the group users of the server. So you can connect with LDAP into ScriptCase and you have the LDAP privileges

there.

This option is available on the left side menu, if you select the LDAP option in **Security Type**

Server

IP of the LDAP server (Needs to be setup before proceeding).

Suffix

Domain of the server.

DN

Domain Component.

User

Administrator of the Server.

Password

User password informed.

Port

Server Port.

After saving the LDAP server connection, It lists the users and groups in the tabs according to the server. Initially, only the administrator user (The same used for LDAP connection) has access to ScriptCase. The Adminsitrator must enable other users and associate them with the groups.

Users

You can see this option when selecting Users in the Security Level. It lists all the users from the LDAP server. We can manage the privileges of each user to the projects and options of ScriptCase.

Groups

You can see this option when selecting Group in the Security Level. It lists all the groups from the LDAP server. You may access this tab to enable the groups and set their privileges.

User / Groups

You can see this option when selecting the security level User/Groups. Here, there is two tabs. In this case, the permissions are defined for the group of users and/or individually by user. On this type of security, you must add the permissions of the group and user.

Users

User Management

This page allows to manage the Users of your Scriptcase. There is a list of all registered users, where we can edit or delete each one. We can also include new users.

This option is available on the left menu when selecting the **Security Type**

User

The username of the registered users.

Insert

Date and time of the user creation.

Applications

It shows how many applications each user has created.

Action

Detail

It shows a detailed view of the user. The email, privileges, projects and his groups.

Edit

It allows changing the privileges and the projects for an user.

Change Password

Change the password of the user.

Let’s see some options.

New User

When clicking on the button **New User** it redirects to a page to inform the user login, password and privileges.

User

The login of the new user.

e-mail

The user e-mail.

New Password

The password of the user.

Confirm Password

Type the password again to confirm it.

Privileges

Administration

The grant to access the main settings area of ScriptCase.

Projects

Let you set which actions the user can do when accessing a project.

Application

The grant to create applications.

Database

It sets the database actions an user can do in the projects.

Project

Set the projects that the user can access.

Then you must select which projects the user can manage.

The project manaager can change the preferences of the project, the existing connections and default values.

Editing a User

The editing page is similar to the new user screen. It has the same options.

User

The login of the new user.

e-mail

The user e-mail.

Privileges

Administration

The grant to access the main settings area of ScriptCase.

Projects

Let you set which actions the user can do when accessing a project.

Application

The grant to create applications.

Database

It sets the database actions an user can do in the projects.

Project

Set the projects that the user can access.

Then you must select which projects the user can manage.

The project manaager can change the preferences of the project, the existing connections and default values.

Changing the Password

Just inform the new password for the user.

Delete Users

We can delete more than one user at the same time

When deleting a user that has applications linked to its login, you must inform if you want to delete the applications or if you want to associate them with another user.

Delete Application

It deletes all applications created by the user.

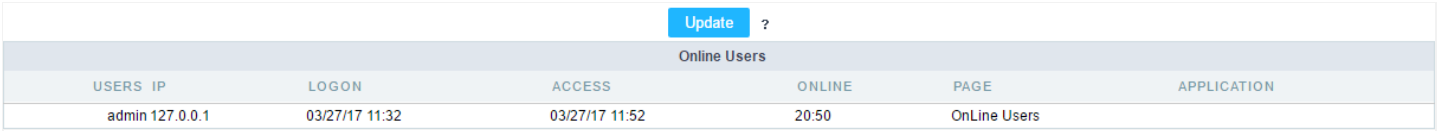
Move Application

It associate the applications to another user.

Services

Online

This page shows all logged users into Scriptcase in real time. The current page each user is accessing, the date and time of access and other information.



Users

Username of the online user.

IP

IP of the workstation where the user is accessing.

Logon

Date and time of log in.

Access

Date and Time of the last know user activity in session.

Online

Time of the user current session.

Page

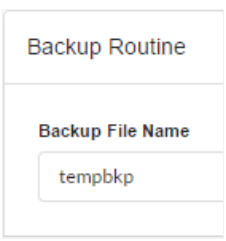
The page is the user accessing at the moment.

Application

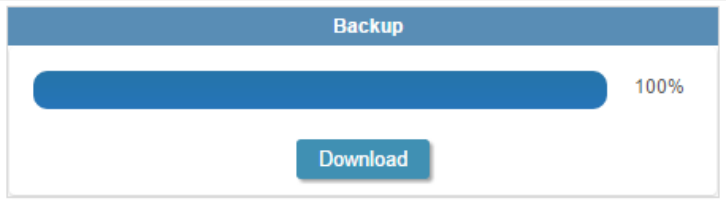
The Application is the user accessing at the moment. This information may return empty if the user is accessing the ScriptCase settings.

Backup

ScriptCase's backup works similarly as the project export, but the backup does the export of all applications of your ScriptCase. To perform a backup, inform the name to generate the file.

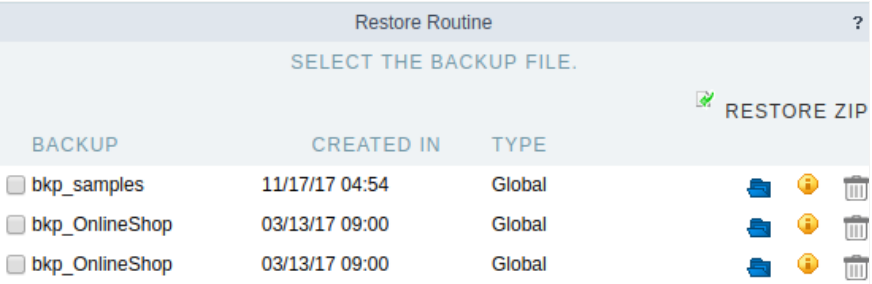


Finally, download the generated file and save it in a secure location.



Restore

This option is similar to Project Import, however using the restore you can import more than one project at a time. When you access the restore you will see a list of backups performed recently, you will be able to restore these backups or to import a backup exported from other Scriptcase installation.



Backup

Backup generated files.

Created in

Creation date and time of the listed backups.

Restore

Backup restore.

Detail

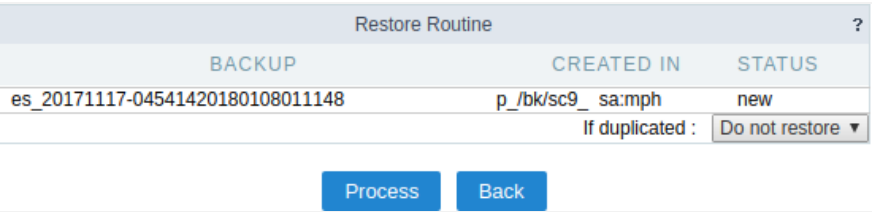
Backup details.

Delete

Deletes the backup permanently.

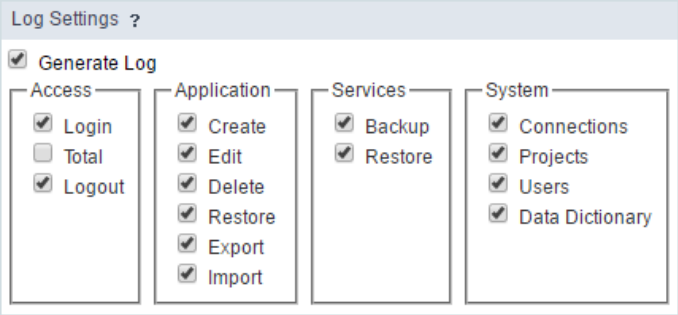
To restore another backup, which is not in the list, select **zip restore** and upload a backup file created by Scriptcase.

After choosing the backup file, confirm the restoration. You must inform if you want to overwrite duplicate files or keep the existing ones in Scriptcase.



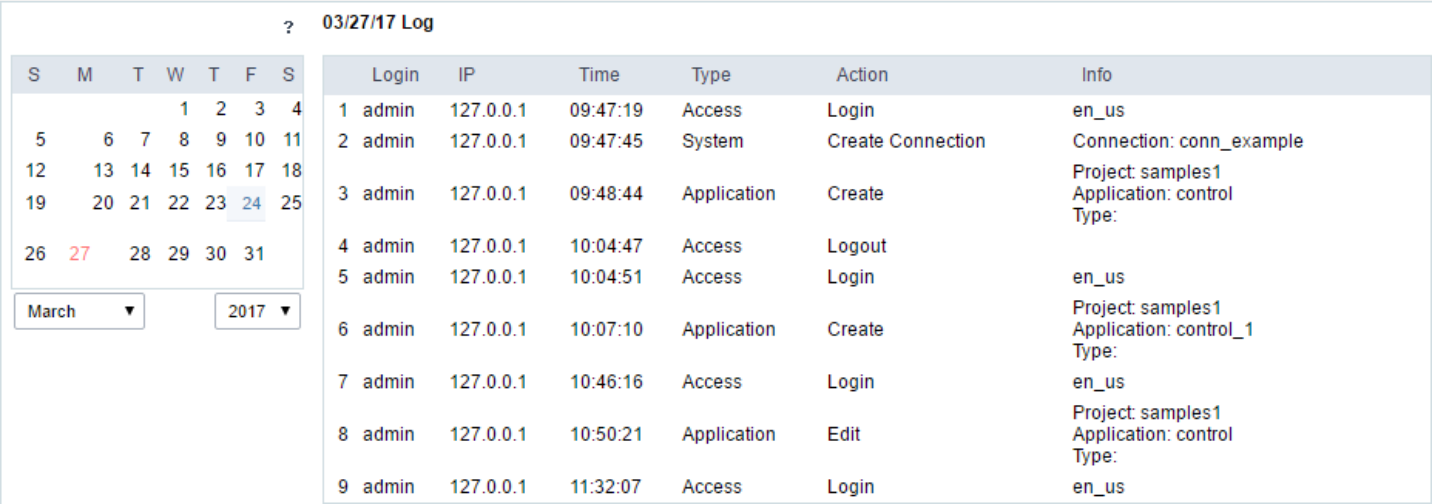
Configure Log

ScriptCase stores (in the database) a log of the main actions done by the users. We can customize the log registry by selecting or not the actions that we want to store.



View Log

This option allows you to view the main actions done by the users separately by day. You can view the information of a desired date by clicking on the calendar. You can distinguish the days that have log with a blue background.



Login

Login of the user responsible for the action.

IP

IP of the computer that the user did the actions.

Time

Time when the action was done.

Type

The type of action done.

Action

What action was done.

Info

Details of the action;

Scriptcase Update

ScriptCase update area. All the update process is done by the INTERNET, using our servers. We recommend using a fast connection.

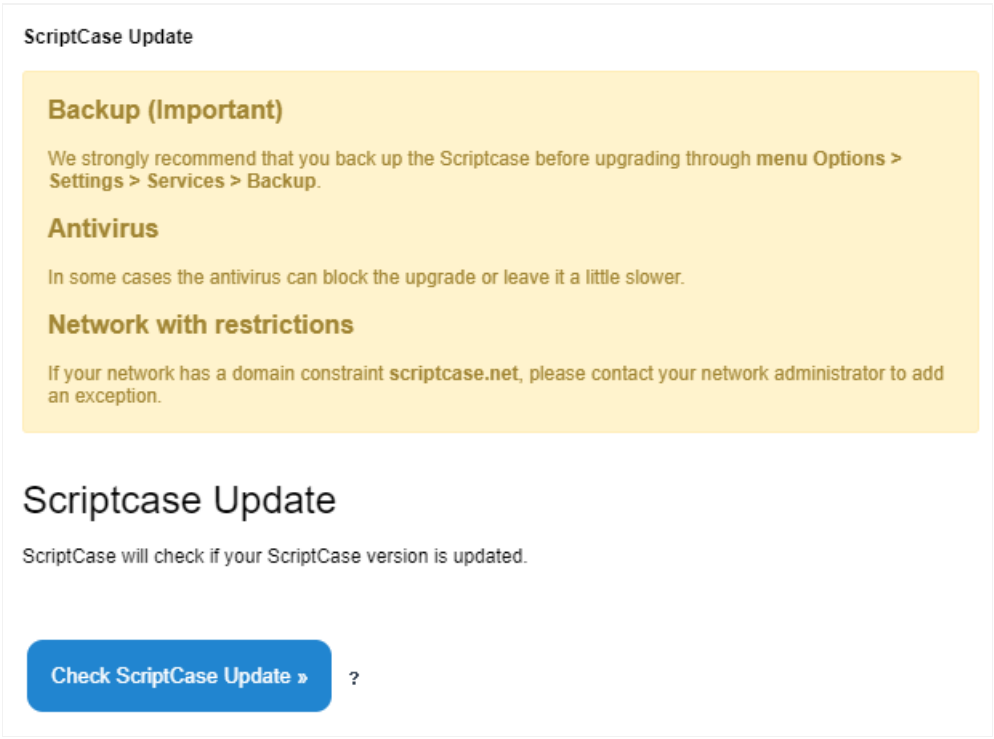
If the computer hasn't an INTERNET connection, it is necessary to realize the Scriptcase manual update.

Check our documentation about **Manual Update** according to your operating system:

- Windows
- Linux
- MacOS

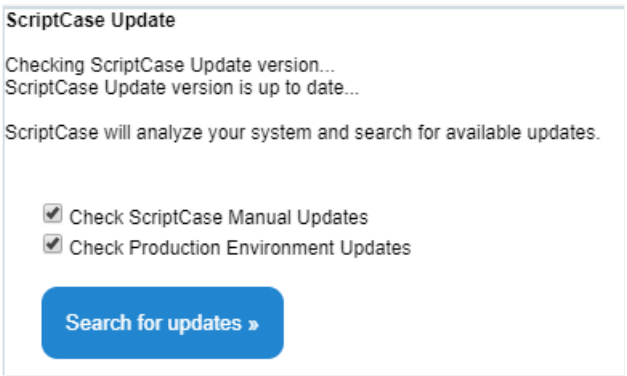
Verifying Updates

Firstly, we must check if a new version of ScriptCase is available.



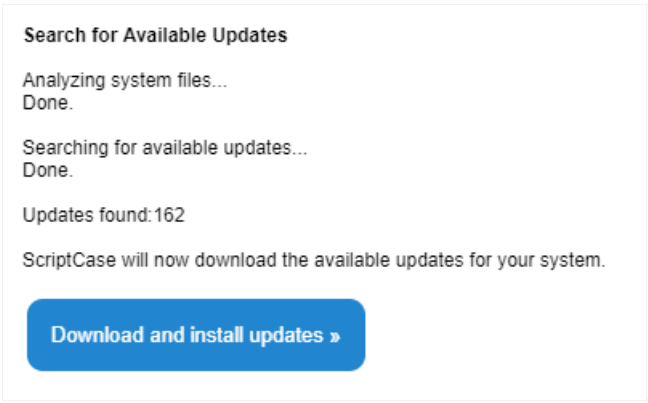
Selecting Items for Update

Then we can choose if we want updating the documentation and "prod" files.



Applying Scriptcase Updates

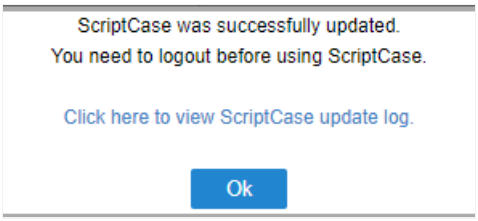
Then, we can see the total of files update found.



Do not interrupt the process after start the Update.

Ending the Update ProcessS

After downloading and updating all the files, you can see the log of changes done in the process: `Click here to view ScriptCase update log` .



If you are having problems connecting to our server, check if your firewall is allowing the connections to these links:

- **scriptcase.com.br**
- **scriptcase.net**

Online Registration

The **Online Registration** is made on a quick way. In order to do this kind of registration the machine where the scriptcase is installed must have internet access.

> In addition to internet access, the network must have permission to access the domain scriptcase.net,for communication to take place.

Online Registration

On this screen, we must inform the data necessary for the registration, **User** and **Password** (Login and Passowrd to access the site) and the **Serial** linked to the informed account.

> The data used for the Scriptcase registration (Username and Password) refer to the user who owns the license (account of the site used to acquire a tool license). The login and password used to access Scriptcase are only for access to the development environment

User

Inform the user who owns the license. This **usuary** is the same **used** to perform **login on our site**,and access the client area.

Paasword

Enter the password of the user informed on the field above. This password is the same used to login in our website.

> If you have forgotten your password to access the website, click on the link to recover your password: https://www.scriptcase.net/password_retrieval/

Serial

Inform the serial that will be used for registration. This serial must belong to the account informed in the field *User*.

> If you have recently purchased a license, your serial will be available in your purchase confirmation email. You can also view your serials in the client area on our website. See how to find your serial on the site.

Confirmation

After completing the registration, a confirmation screen will appear, like the one below. Click on OK, you will be redirected to login screen.

Offline Request

The *Offline Request* is the first stage of the **Offline Registration**, it is where we will obtain the file used on the registration.

> See a step by step of how to do an Offline registration

Offline License Request

On this screen, we need to inform the data necessary for registering the tool: **User** and **Password** (Login and password to access the site) and a **Serial** linked to the informed account.

After entering the details, click on Request.

> The data used for the Scriptcase registration (Username and Password) refer to the user who owns the license (account of the site used to acquire a tool license). The login and password used to access Scriptcase are only for access to the development environment

User

Inform the user who owns the license. This **usuary** is the same **used** to perform **login on our site**,and access the client area.

Paasword

Enter the password of the user informed on the field above. This password is the same used to login in our website.

> If you have forgotten your password to access the website, click on the link to recover your password: https://www.scriptcase.net/password_retrieval/

Serial

Inform the serial that will be used for registration. This serial must belong to the account informed in the field *User*.

> If you have recently purchased a license, your serial will be available in your purchase confirmation email. You can also view your serials in the client area on our website. See how to find your serial on the site.

Download and validation of the file

On this screen we have the **Download** button, to download the scriptcase_v9.req file with some data necessary for registration and the access link to the registration request pagehttps://www.scriptcase.net/licenserequest/

> We must download the file and place it on a portable storage device so that we can move it to a machine with internet access, since the next step will be to upload this file on our website.

> See a step-by-step of how to do an Offline registration

Form for generating the .lic file (Site page)

The link https://www.scriptcase.net/licenserequest/ send us to the .lic file generation form, where we need to inform the serial number and upload the .req file generated in Scriptcase.

> You must inform the same serial used to generate the .req file and upload the file in the upload field.

A new file will be generated with the lic extension (scriptcase_v9.lic). Download and save the file to a portable storage device.

This file will be used for the second step of Offline Registration.

Offline Registration

This is the final step in the process to make an **Offline Registration** , where the tool registration will actually be performed.

To proceed, you must have the .lic file generated in the step Offline Request.

> The generation of the file carried out in the step **Offline Request** is mandatory for offline registration. See a complete step-by-step of how to do Offline registration

Offline License Registration

On this screen, we need to inform the data necessary for registering the tool: **User**, **Password** (Login and password to access the website) and a **Serial** linked to the informed account. In addition, we need to upload the .lic file previously generated.

> The data informed must be the same used on the step Offline Request where the .req was generated

User

Inform the user who owns the license. This **usuary** is the same **used** to perform **login on our site**,and access the client area.

Paasword

Enter the password of the user informed on the field above. This password is the same used to login in our website.

> If you have forgotten your password to access the website, click on the link to recover your password: https://www.scriptcase.net/password_retrieval/

Serial

Inform the serial that will be used for registration. This serial must belong to the account informed in the field *User*.

> If you have recently purchased a license, your serial will be available in your purchase confirmation email. You can also view your serials in the client area on our website. See how to find your serial on the site.

Upload field

Upload the .lic file generated in the step Offline Request

Confirmation

After clicking on **Register** a confirmation message will be displayed.

> After registration, you will be redirected to the login screen.

My ScriptCase

Interface where you can define the ScriptCase settings. All the users have access to this section, and the changes done are tied to the login(user) that you are currently using. This way each user can have their own settings for ScriptCase.

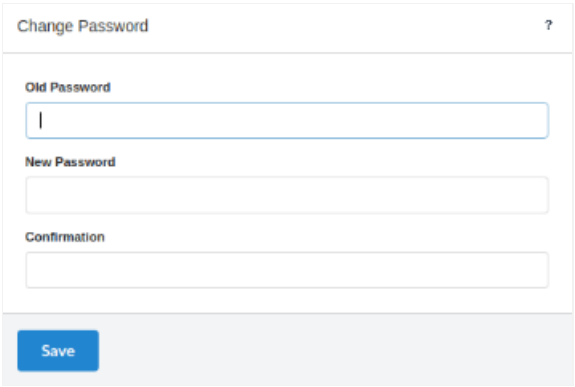
- **Language** - Scriptcase interface language. When informing the language in **My ScriptCase**, the selected language at the login won't be considered.
- **Security Resources** - When Uncheck these options, the security settings won't be considered when running the applications in the development environment. When deploying the applications, they will respect the security settings normally.
- **Initial Information** - Defines the information that will be displayed on the application listing.
- **Show Friendly URL** - Defines the display of the field to inform the Friendly URL in the application.
- **Main Menu Interface** - Defines the width of the left side menu of the home page, where the applications are listed and the option of incrementation.
- **Application Settings Interface** - Defines the width of the left side menu of the applications settings and the option of incrementation.
- **Auto Save** - Allows the modifications to be saved automatically.
- **Display INSERT CODE on events** - Defines if the examples will be displaying in the events of the applications.
- **Convert indentation to tab** - Allows to use "tab" hotkey to indent your code or line selected.
- **Show folders at home** - Allows for the sub-folders to be listed with the applications, when selecting a folder on the side menu in the home of the project.
- **Allow Hotkeys** - This option allows to enable/disable Scriptcase hotkeys. Click Here to view the hotkeys list.

Change Password

Interface where you can change the password of your user. We always recommend using passwords with a certain level of complexity, to prevent hacks of brute force.

Inform your current password, and the new password twice just to confirm it.

For the option of password recovery to work, the administrator need to setup the SMTP configurations correctly in ScriptCase's settings.

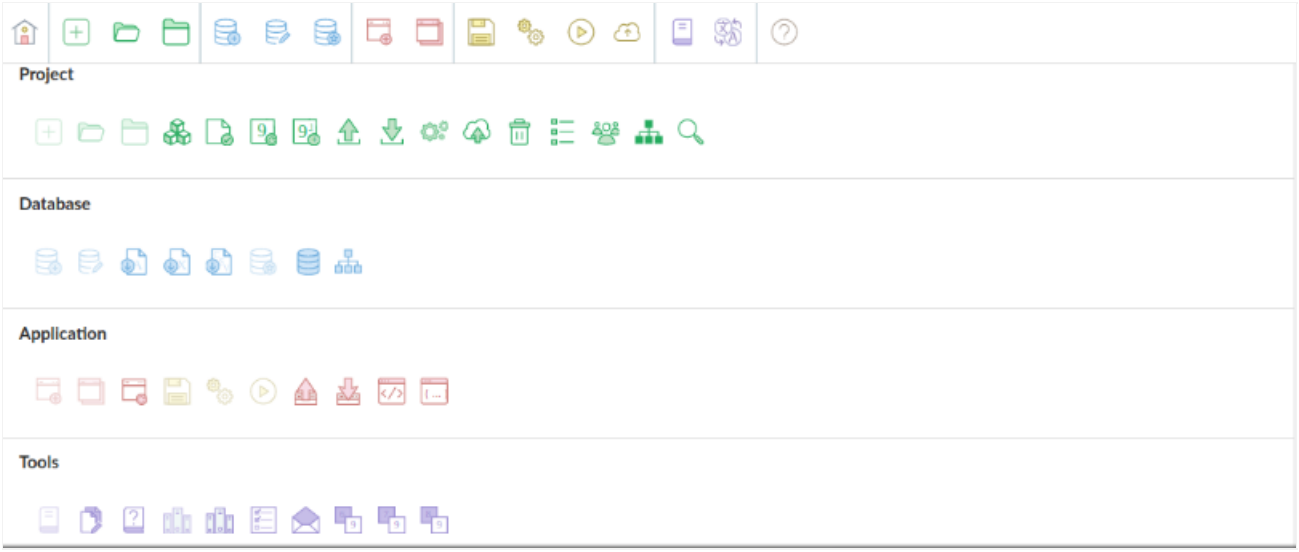


- **Old Password** - Inform your current password.
- **New Password** - Inform your new password.
- **Confirmation** - Confirm the new password.

My Toolbar

Allows each user to customize their toolbar, by including or removing unnecessary items for your usage, giving you a better experience while developing the applications. This option is tied to the user that did the modification, this way each user can modify their own toolbar the way they desire.

The icons are divided in categories. To include an item to the toolbar, we just need to drag it to the toolbar.



- **Toolbar** - Icons that'll be displayed on the toolbar of ScriptCase.

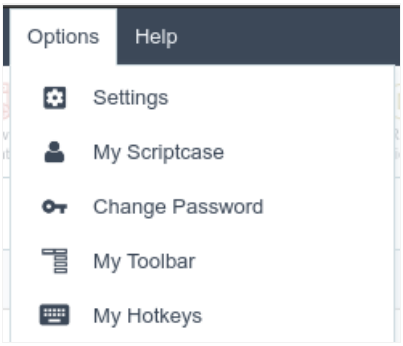
Buttons

- **Save** - Saves the modifications done to the toolbar.
- **Add Separator** - Adds a separator to the toolbar, after the last item of the current setup.
- **Create item** - Creates a new item in the toolbar. Inform the URL, icon and description of the new item.
- **Return to default** - Restores the toolbar to its default configuration.

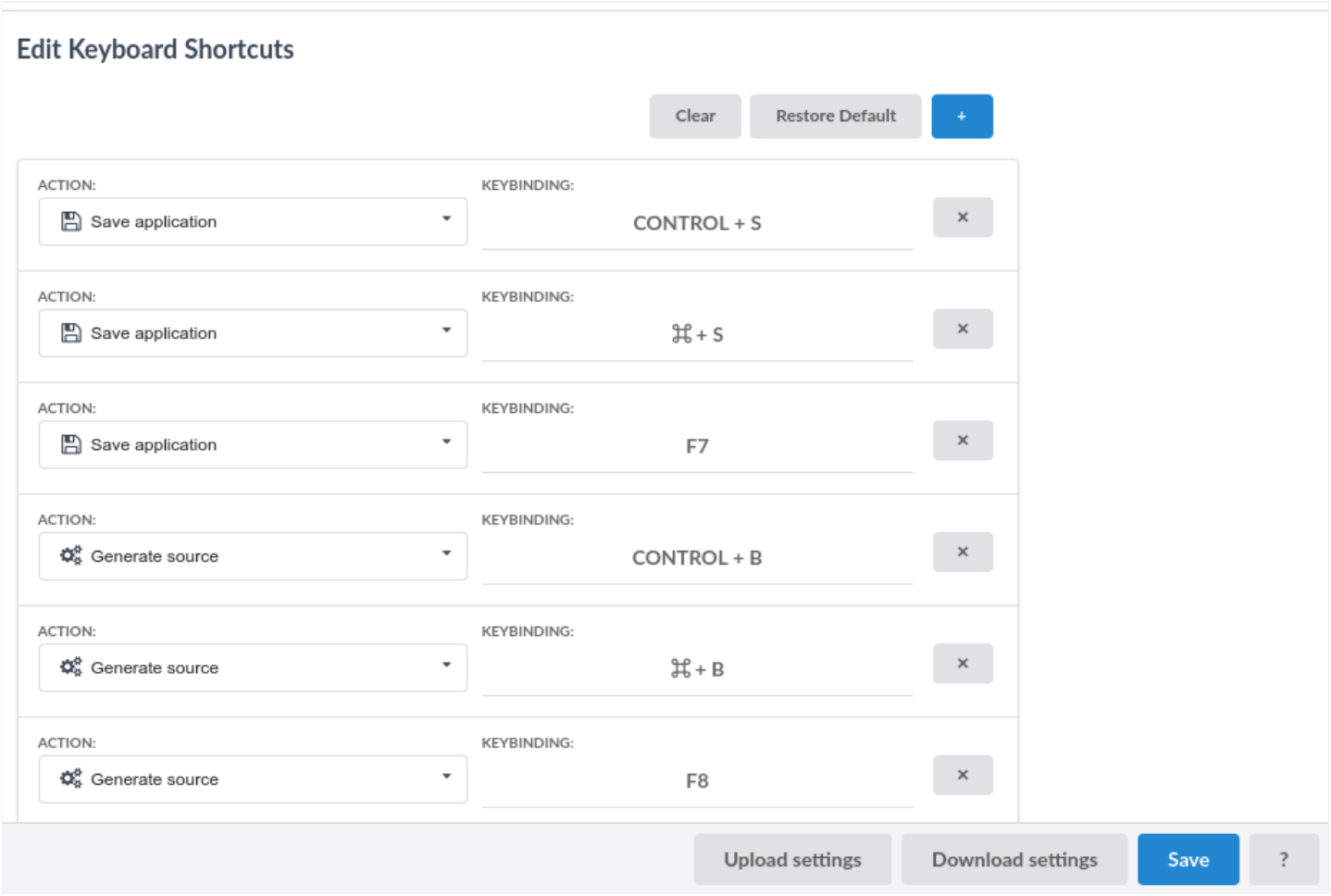
My Hotkeys

Allows you to change the default Hotkeys of your Scriptcase interface, offering to you more practicality and customization to your development.

To change the default hotkeys, you just need access the menu **Options > My Hotkeys**.



Will be open the hotkeys and the setting options, as the image below:



- **Clear**: Clears all predefined settings by Scriptcase or by user.
- **Restore Default**: Restores all the default shortcuts that was changed or erased.
- **Action**: Selects the action that will be made when press the selected keybind.
- **Add "+"**: Adds a new action in the keybind list.
- **Keybinding**: Selects the keybinds that will be responsible for execute the chosen action.
- **Upload Settings**: Realizes the importation of the keybind settings exported.
- **Download Settings**: Realizes the download of the keybind settings in use.

Deployment Overview

Publishing is the process of deploying applications/projects developed in Scriptcase to be accessed by end-users. This process involves generating the project's files and configuring the production environment.

In Scriptcase, there are two types of publishing.

Typical Publishing {id-01}

A simplified and automated publishing process. In this type of publishing, the configurations of the **production environment directories** and the **project connection names** are set by default. Thus, all the necessary files for the system's operation are placed inside the **_lib** folder when generating a publication.

To proceed with the Typical Deploy Click Here.

Production Environment Directories in Typical Publishing

- Production environment: `\_lib\prod`
- Image directory: `\_lib\file\img`
- Tmp folder: `\_lib\tmp`
- File directory: `\_lib\file\doc`

Advanced Publishing {id-02}

This type of publishing allows the developer to customize the location of the production environment files on the server, enabling, for example, the use of the same production environment for several projects published on the same server.

During the publishing process, the developer must specify the **production environment directories** and **link the connection names** from the development environment to the production environment. Therefore, it is recommended that the developer have a production environment with the file structure ready, so that during publishing, the correct file locations can be specified.

To proceed with the Advanced Deploy Click Here.

Prerequisites for Publishing

For applications to function correctly outside the development environment, some requirements must be followed.

- **Web Server**: Have a configured web server: Apache (**recommended**), IIS, nginx, or another;
- **Database**: Enable the database driver that will be used by the system. Check how to enable the driver for your database.
- **Production Environment**: Keep the production environment updated and compatible with the current version of the tool;
- **Necessary PHP Functions**: Some PHP functions must be enabled for the correct functioning of some application features. Check the list of these functions
- **Compatible PHP**: Have a PHP configured on the web server, compatible with the version of Scriptcase used. Check the compatibility table below.

PHP Compatibility in Development with PHP in Production

The PHP version used in the development environment directly affects the PHP version that can be used in the production environment.

See the table below.

PHP of Installation (Development)	Compatible PHP in Production Environment (Production)				
Scriptcase with PHP 7.3	PHP 7.2	PHP 7.3	PHP 7.4		
Scriptcase with PHP 8.1	PHP 7.2	PHP 7.3	PHP 7.4	PHP 8.0	PHP 8.1

Typical Deploy

Pre-Requirements

For applications to function correctly outside the development environment, some requirements must be followed.

- **Web Server**: Have a configured web server: Apache (**recommended**), IIS, nginx, or another;
- **Database**: Enable the database driver that will be used by the system. Check how to enable the driver for your database.
- **Production Environment**: Keep the production environment updated and compatible with the current version of the tool;
- **Necessary PHP Functions**: Some PHP functions must be enabled for the correct functioning of some application features. Check the list of these functions
- **Compatible PHP**: Have a PHP configured on the web server, compatible with the version of Scriptcase used. Check the compatibility table below.

PHP Compatibility in Development with PHP in Production

The PHP version used in the development environment directly affects the PHP version that can be used in the production environment.

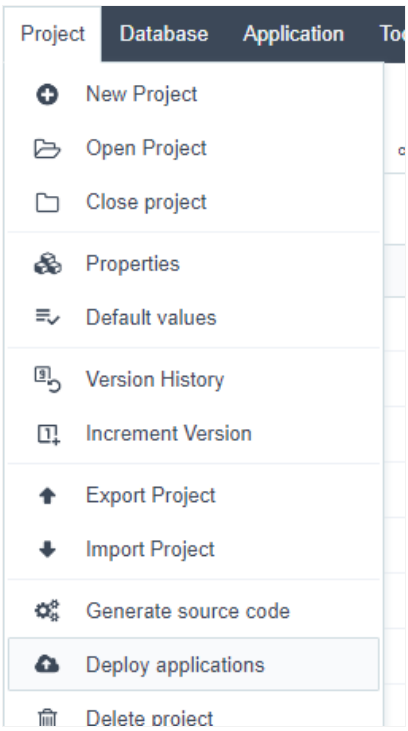
See the table below.

PHP of Installation (Development)	Compatible PHP in Production Environment (Production)				
Scriptcase with PHP 7.3	PHP 7.2	PHP 7.3	PHP 7.4		
Scriptcase with PHP 8.1	PHP 7.2	PHP 7.3	PHP 7.4	PHP 8.0	PHP 8.1

This type of deployment is for users that want the Production Environment (Common Libraries) with the default settings from ScriptCase, the basic setup is done already be ScriptCase. We will show the entire process of deployment in this documentation.

Project Deployment

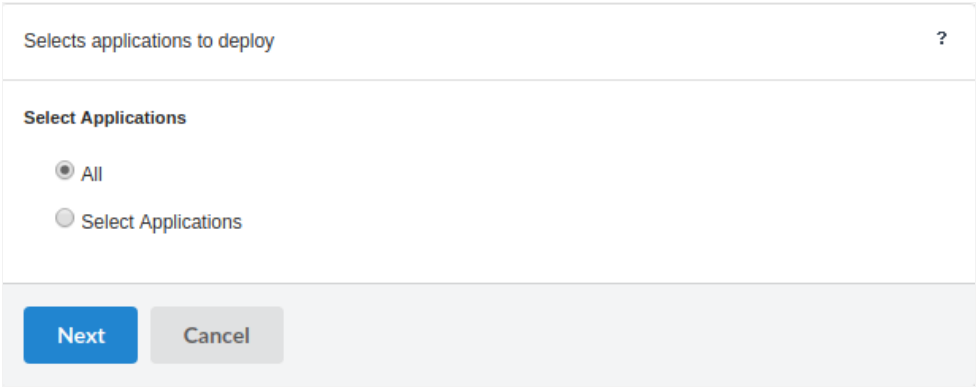
Access your project and choose the option **“Project -> Deploy Project”**, available on the top menu.



After clicking on the Deploy applications button, you will have the option to deploy all the applications from the project or you can select the applications that you want to deploy.

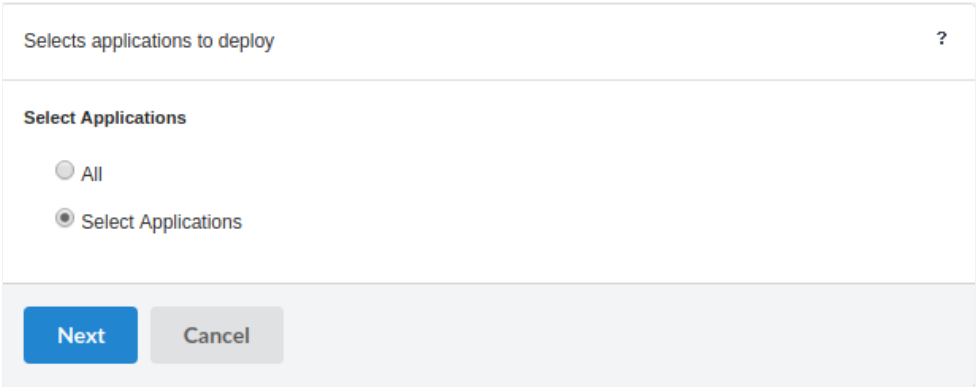
Select All Applications

When select this option, all the applications of the project will be deployed.



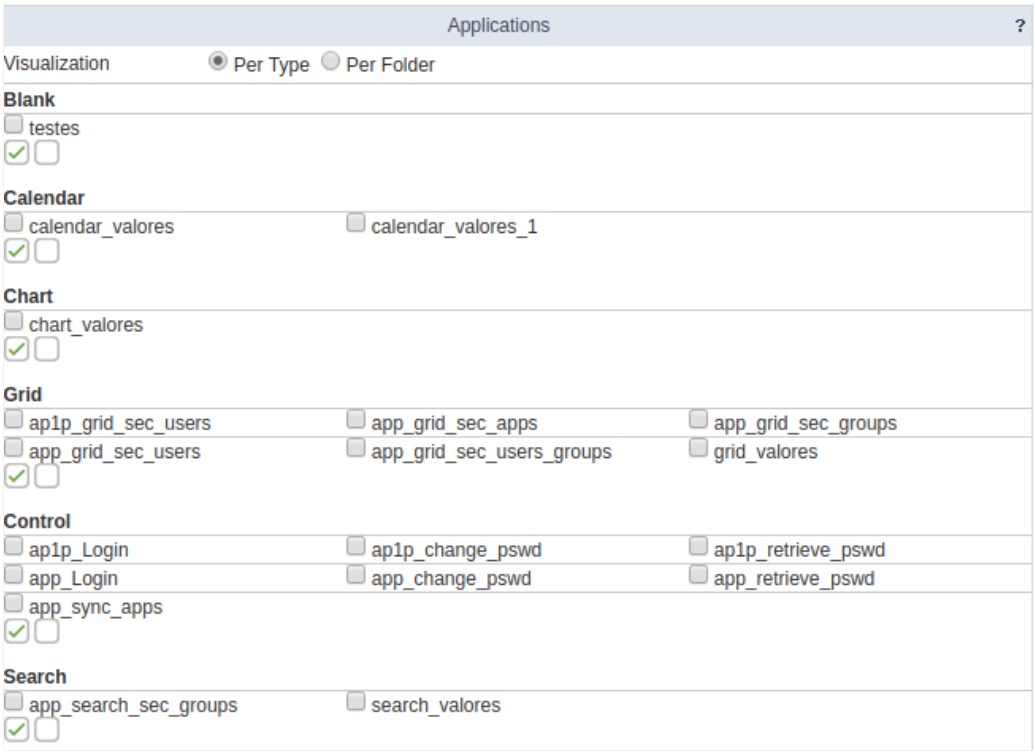
Select the applications individually

This option allow us to select the applications that will be deployed.



List of available applications

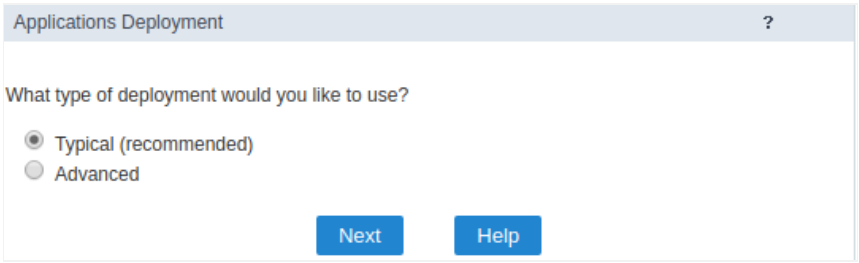
During this process, the Scriptcase list all the Applications in the project so you can select the applications that you wish to deploy.



Option	Description
Per type	Selecting this option, you can see the list of applications grouped by your respective type.
Per folder	Selecting this option, you can see the list of applications grouped by your respective folder.

Choose of the deployment type (Typical)

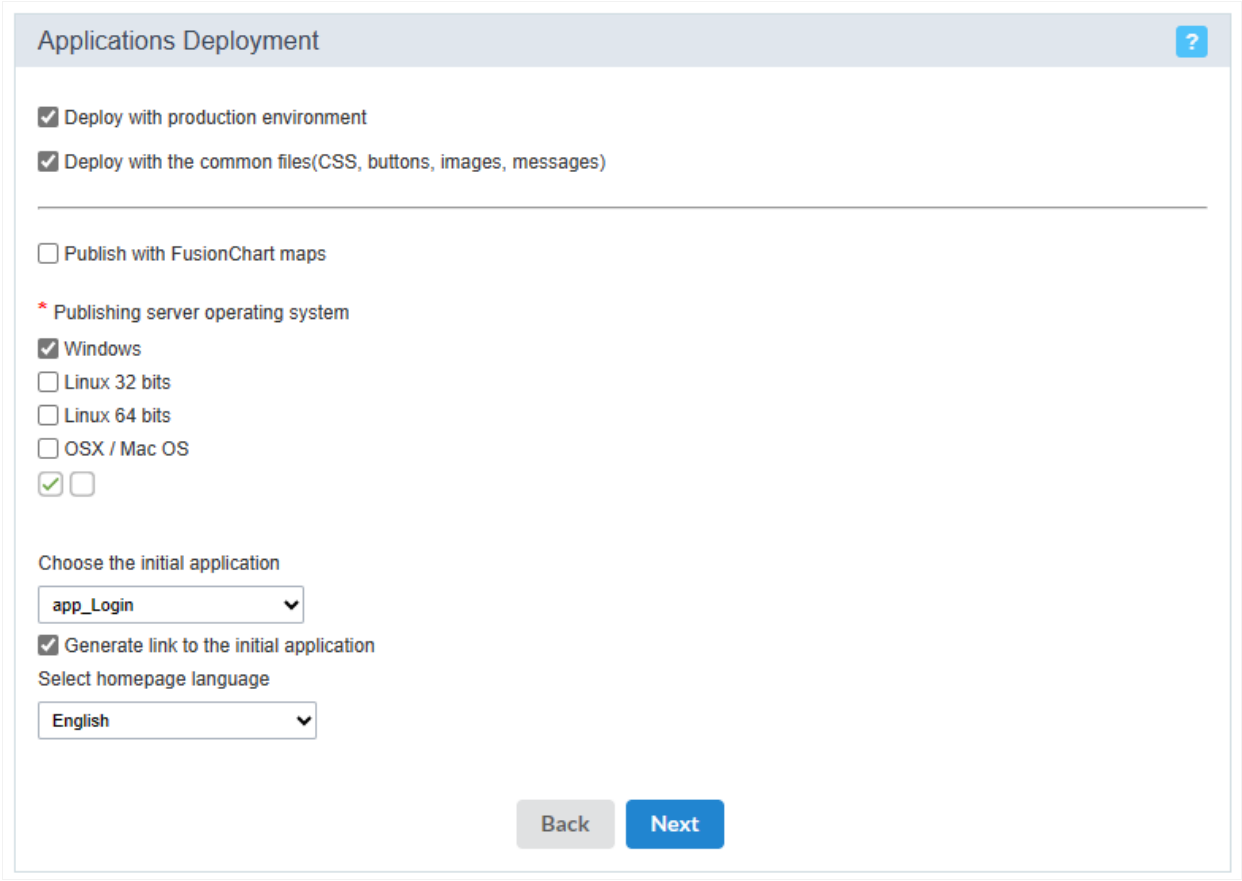
Next, we need to choose the type of desired deployment.



Properties of a deployment

Deployment properties

We must define which application will be the deployment initial application. In this step, we also define if we will send the common libraries with our project.



It is necessary to select the common libraries, which are the files responsible for the Production Environment. It is necessary to select the common files, for the images, css and custom messages to be sent at the moment of the deploy.

Option	Description
Deploy with the Common Libraries(Prod)	Contains the used libraries by our applications and also the informations of connections used by the project.
Deploy with the common files	This option define if the files that make up an application will be sent in the publication files. The files are: Images, css and customized messages.
Publish with FusionChart Maps	Checking this option, the referred files to the FusionCharts library will be added in the deployment file.
Publishing server operating system	This is a mandatory option for a project deployment, when you select the operating system, the Scriptcase will create the correct structure for your deployment.
Choose the Initial Application	Defines the initial system application.
Generate link to the Initial application	This feature avoids the creation of redirections in the production environment and creates an access screen for the initial application defined in the previous attribute. This configuration should prevent the system from being mistakenly identified as an insecure page by some browsers.
Select homepage language	Defines the language of the text that will be displayed on the home page, where the button for accessing the home application will be displayed.

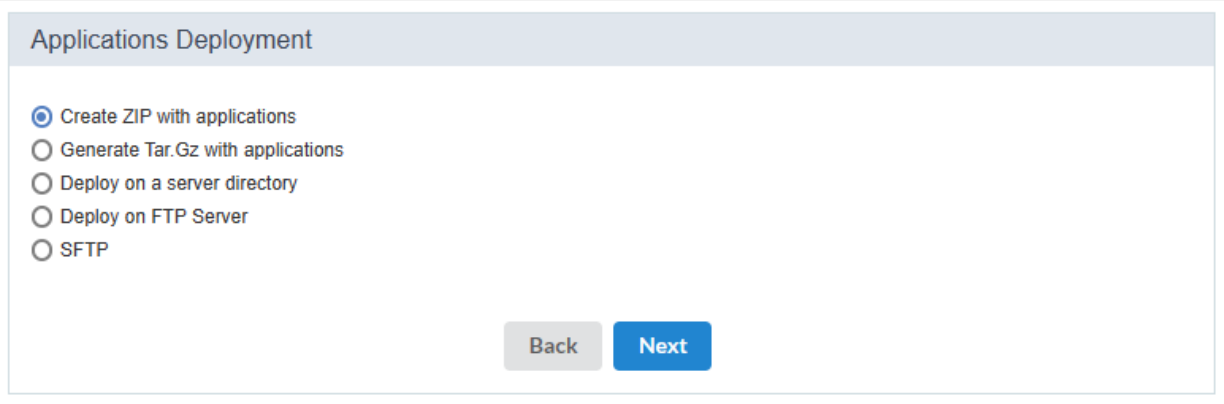
Generated Files

At this stage, the way in which the publication will be carried out will be defined.

- **Generate ZIP with applications** - In this option, at the end of the publication, a button will be displayed to download the .zip file generated with the publication files. We recommend using it to publish in production environments configured on Windows.
- **Generate Tar.Gz with applications** - A .Tar.Gz will be generated with the publication files, at the end of the process the file download will be displayed. This type of compression is recommended for publishing in production environments configured on Linux or MacOS.
- **Publish to directory on the server** - Allows direct publishing to a local directory, where Scriptcase is installed. This type of publication is recommended for servers that share the development and production environment. The publication files will be generated within the indicated folder, therefore, we recommend using an exclusive folder for publication. For example, when indicating the C:\Program Files\NetMake\v9-php81\wwwroot\project_name directory, the application files, and the _lib directory with css and themes will be placed directly in the project_name folder.
- **Publish to FTP server** - Sends the publication files directly to the configured FTP server. The publication files will be placed in the indicated directory, therefore, we recommend using a specific directory for the publication.
- **SFTP** - Just like FTP, publishing via SFTP sends the publication files directly to the configured server, placing the publication files in the specified directory.

Generate ZIP or Tar.Gz with applications

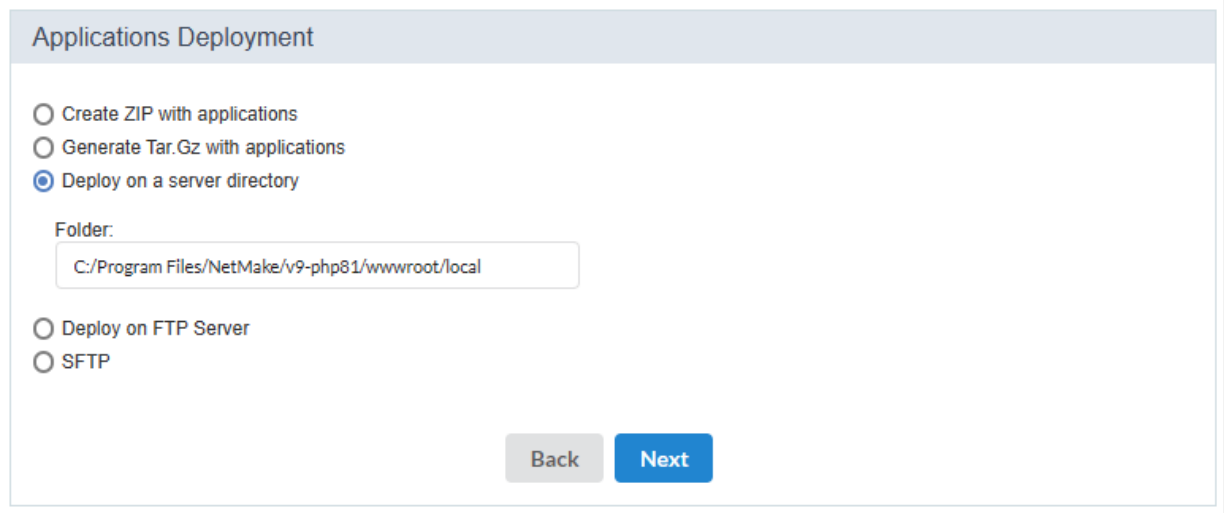
The Scriptcase will generate a file **Zip** or **Tar.Gz**. In the end of this process, will be displayed a report of the deployed applications with the download link for download the project.



The file **ZIP** or **Tar.Gz** generated, must be decompressed only in the deployment server, to avoid that the files be corrupted.

Deploy on a server directory

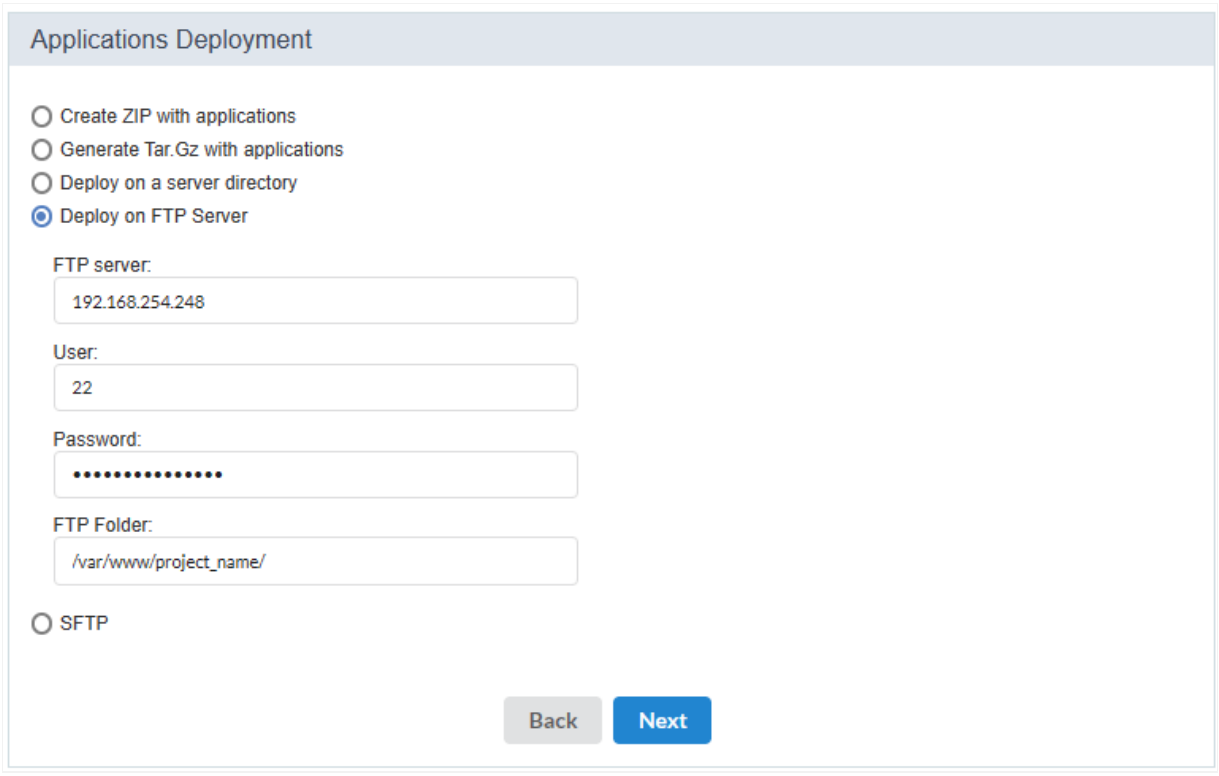
Deploys the project in the informed directory. We must inform the absolute path to the deployment directory.



This option should be used in the case that the deploy is going to be in the same sever where ScriptCase is installed, otherwise this option is not valid.

Deploy on FTP server

This option generates the files directly in the FTP remote server.

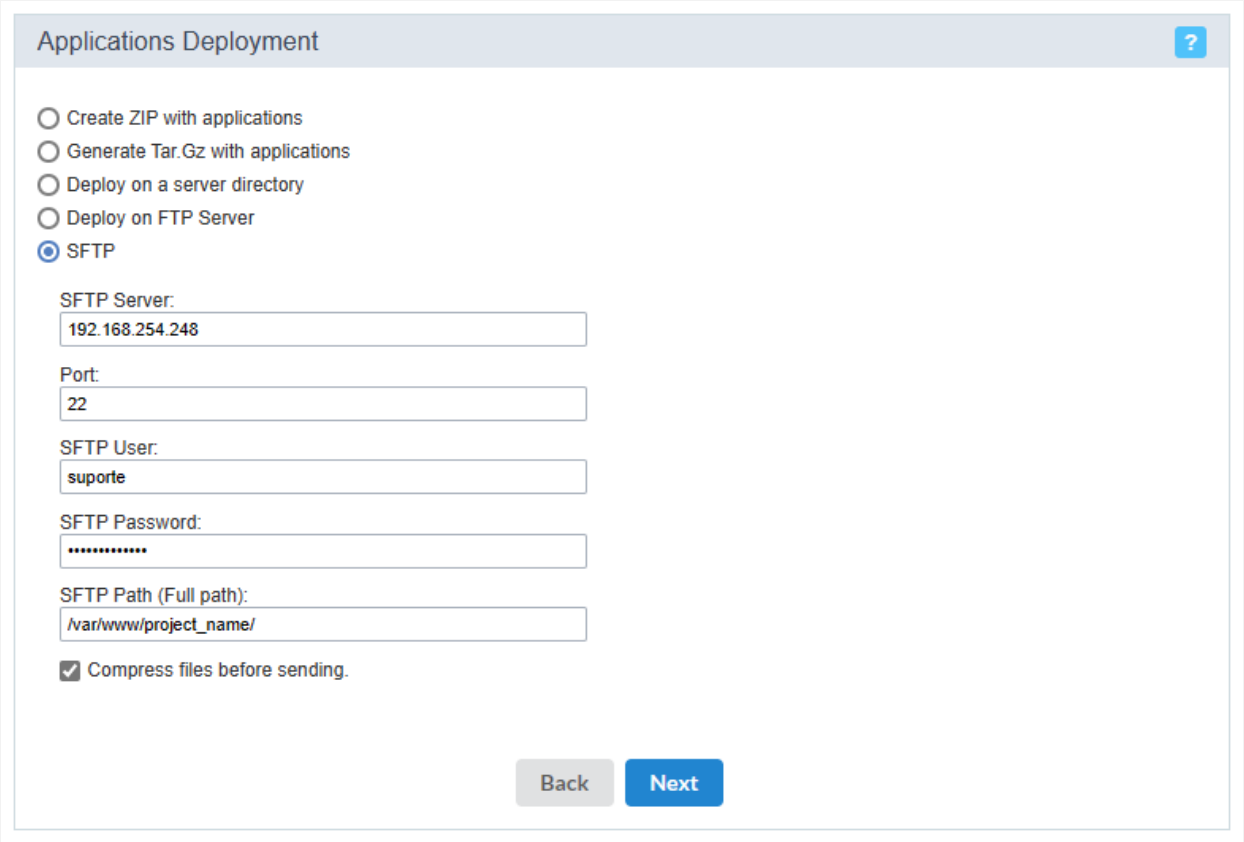


Options	Description
FTP server	Server where you will deploy your project. Inform the domain or the server's IP.
User	User for the FTP server in the production, remembering that the user need to have permission to write in the server folder, otherwise it will not work.
Password	Password for the FTP user.
FTP Folder	Server directory where the project will be deployed in.

To use this option is necessary that you have a server that provides the FTP access and an user with write permissions.

Deploy on a SFTP server

Deploys the application automatically on a remote SFTP server.

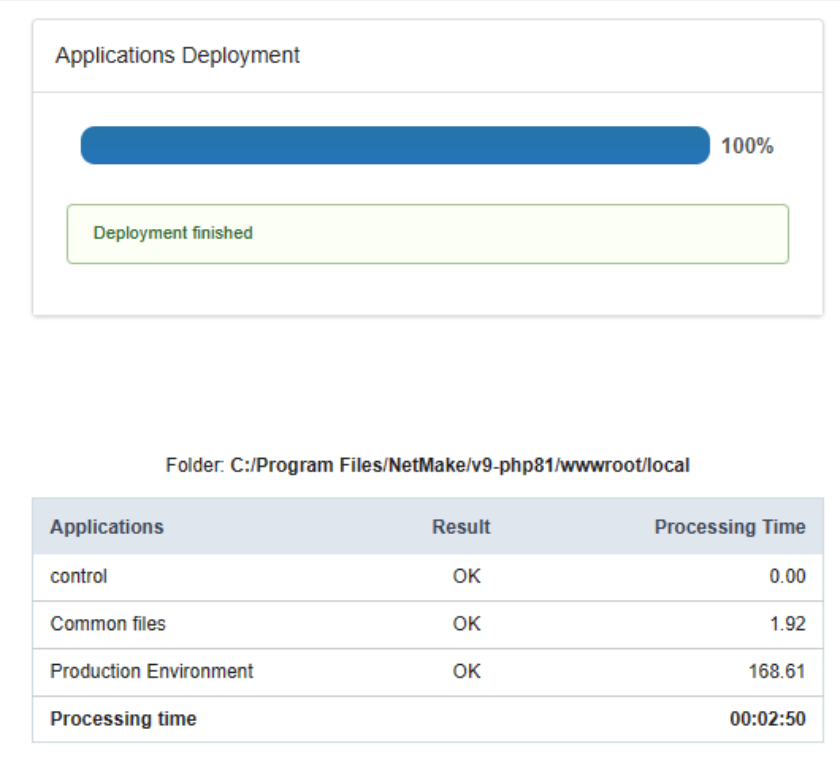


Options	Description
SFTP server	Server where you will deploy your project. Inform the domain or the server's IP.
Port	Defines the port that will be used when connecting to the server.
User	User for the SFTP server in the production, remembering that the user need to have permission to write in the server folder, otherwise it will not work.
Password	Password for the SFTP user.
SFTP Folder	Server directory where the project will be deployed in.
Compress files before sending	This option defines if the files will be compressed before being sent to the server or if they will be sent one by one. We recommend checking the option so that the files are compressed before sending, in this way the publication tends to be much faster, especially when publishing with a large number of files.

To use the **SFTP** option is necessary that the user have permission to write on the server folder.

Deployment log

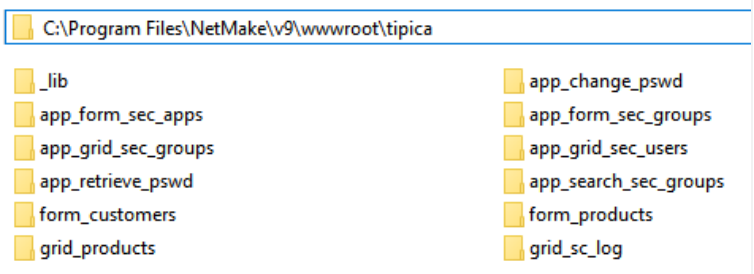
At the end of the process you will receive a report with all the applications deployed and a download link of the project ready to be published.



Applications	Result	Processing Time
control	OK	0.00
Common files	OK	1.92
Production Environment	OK	168.61
Processing time		**00:02:50**

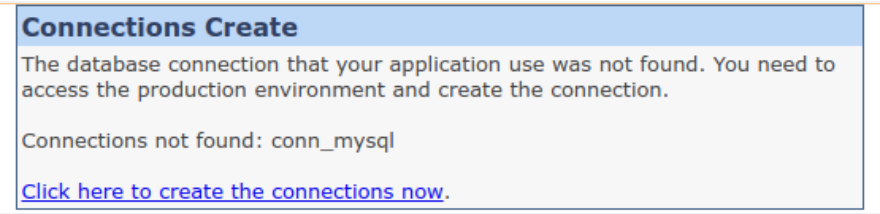
Server Directories

In the server root was created a folder(typical) where we will decompress the deployment files.



In the end of the compression process, we must access your project through the browser.

In the first access we must configure the connections in Production Environment.



Advanced Deploy

Pre-Requirements

For applications to function correctly outside the development environment, some requirements must be followed.

- **Web Server**: Have a configured web server: Apache (**recommended**), IIS, nginx, or another;
- **Database**: Enable the database driver that will be used by the system. Check how to enable the driver for your database.
- **Production Environment**: Keep the production environment updated and compatible with the current version of the tool;
- **Necessary PHP Functions**: Some PHP functions must be enabled for the correct functioning of some application features. Check the list of these functions
- **Compatible PHP**: Have a PHP configured on the web server, compatible with the version of Scriptcase used. Check the compatibility table below.

PHP Compatibility in Development with PHP in Production

The PHP version used in the development environment directly affects the PHP version that can be used in the production environment.

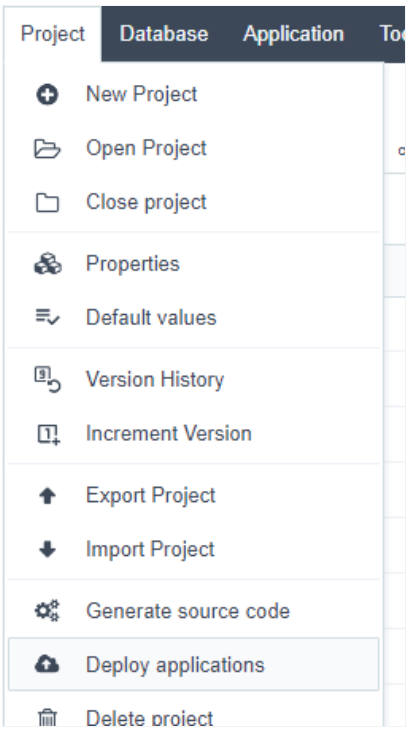
See the table below.

PHP of Installation (Development)	Compatible PHP in Production Environment (Production)				
Scriptcase with PHP 7.3	PHP 7.2	PHP 7.3	PHP 7.4		
Scriptcase with PHP 8.1	PHP 7.2	PHP 7.3	PHP 7.4	PHP 8.0	PHP 8.1

Different from the typical deploy, where the directories' setup is done from ScriptCase using the default settings, the advanced deployment allows you to choose the directories is according to the server's architecture where it is going to be deployed.

Project Deployment

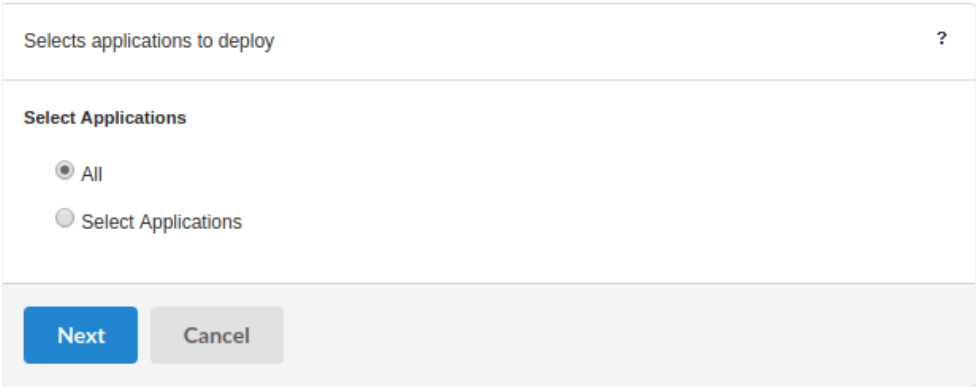
Access your project and choose the option **"Project -> Deploy Project"**, available on the top menu.



After clicking on the Deploy applications button, you will have the option to deploy all the applications from the project or you can select the applications that you want to deploy.

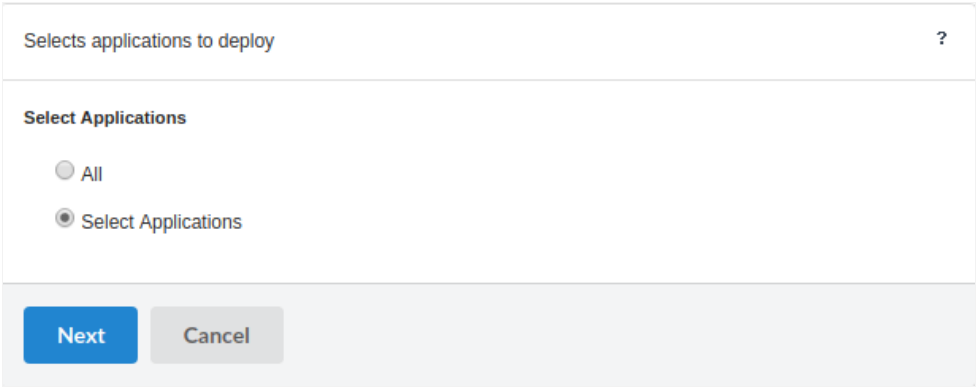
Select All Applications

When select this option, all the applications of the project will be deployed.



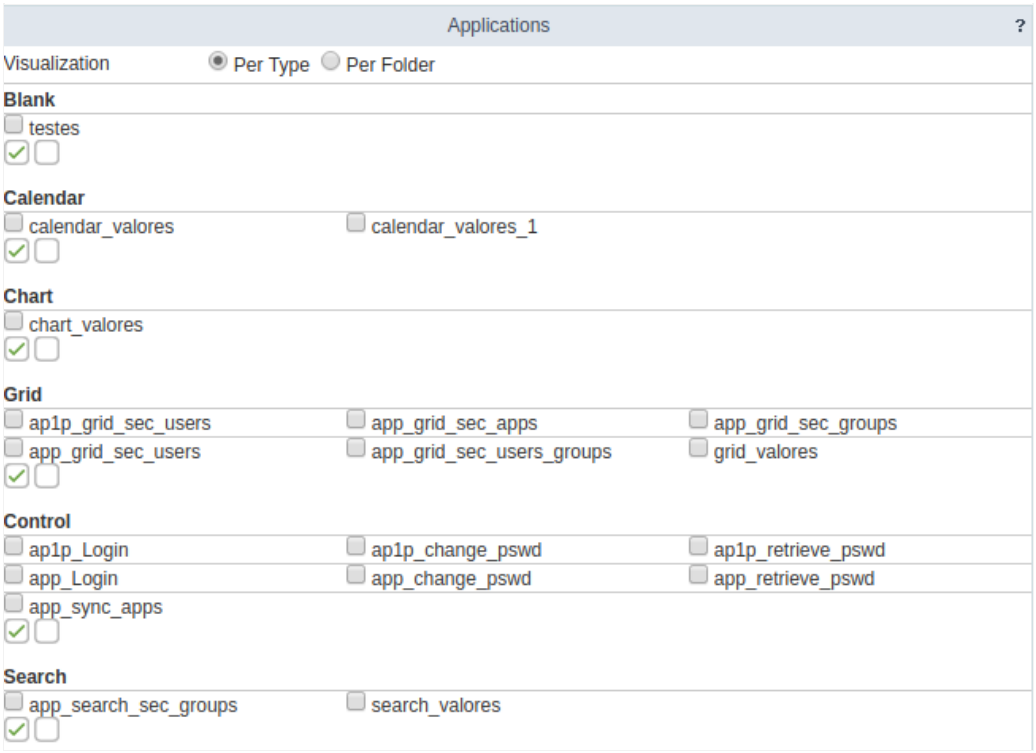
Select the applications individually

This option allow us to select the applications that will be deployed.



List of available applications

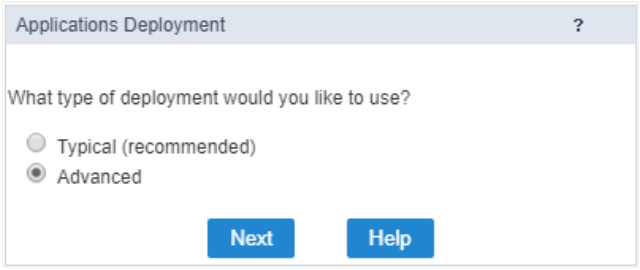
During this process, the Scriptcase list all the Applications in the project so you can select the applications that you wish to deploy.



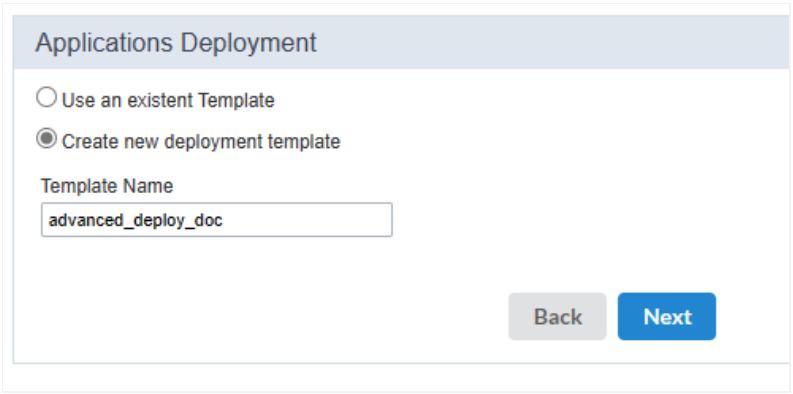
Option	Description
Per type	Selecting this option, you can see the list of applications grouped by your respective type.
Per folder	Selecting this option, you can see the list of applications grouped by your respective folder.

Choose of the deployment type (Advanced)

Next, we need to choose the type of desired deployment.



Create a deployment template to save the defined settings. Or, you can use an existing template to reuse the established settings in your project.

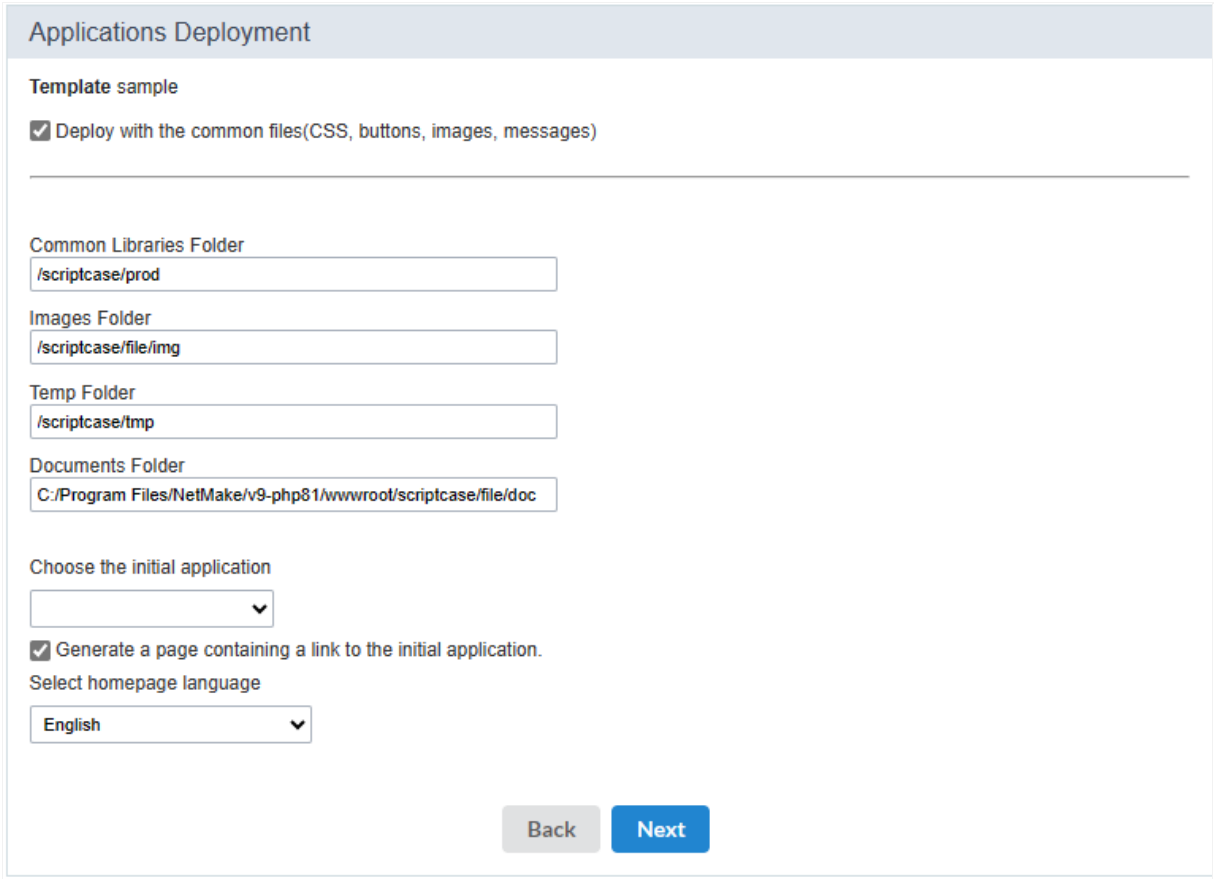


Template Name

This option is necessary to inform the deployment template name that will be used in the project. This option can be reused in future deployments.

Setting the Advanced Deploy

We must define which application will be the project initial application. In this step we also define if we will send the FusionCharts Maps in the deployment file.

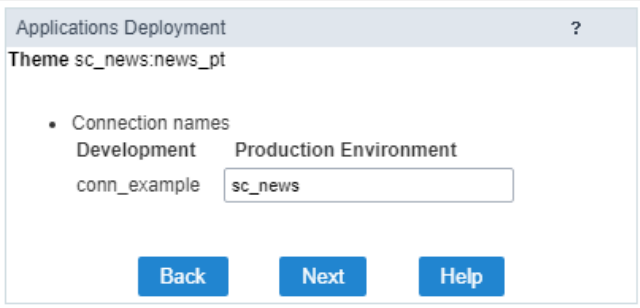


NOTE: It is necessary to select correctly the deployment server operating system, allowing the correct generation of the folders structures for deployment.

Option	Description
Deploy with the common files	This option is **obligatory** for a project deployment, for the images, CSS and customized messages be send in the moment of deploy generation.
Publish with FusionChart Maps	Checking this option, the referred files to the FusionCharts library will be added in the deployment file.
Publishing server operating system	This option is **obligatory** for a project deployment, when you select the operating system, the Scriptcase will create the correct structure for your deployment.
Common Libraries Folder	It is necessary define the main directory where will be deployed the production environment. By default, the Scriptcase recommends to use: `/scriptcase/prod` .
Images Folder	It is necessary define the directory that will be stored the images in the production environment. By default, the Scriptcase recommends to use: `/scriptcase/file/img` .
Temp Folder	It is necessary define the directory that will be stored the temporary files in the production environment. By default, the Scriptcase recommends to use: `/scriptcase/tmp` .
Documents Folder	It is necessary define the directory that will be stored the documents in the production environment. By default, the Scriptcase recommends to use: `/home/$user/public_html/scriptcase/file/doc` . **Is this option is necessary to inform the absolute path to the documents folder.**
Choose the Initial Application	Defines the system initial application.
Generate link to the Initial application	This resource prevents the creation of a redirect in the production environment and creates an access screen for the initial application defined in the previous attribute. This configuration should prevent the erroneous identification of the system as an insecure page performed by some browsers.
Select homepage language	Defines the language of the text that will be displayed on the home page, where the button for accessing the home application will be displayed.

Defining the production connection name

Define the connection name that will be used by the project after deploying.



Connection names

In this option, we will inform the names of the connections existents in your project for deployment.

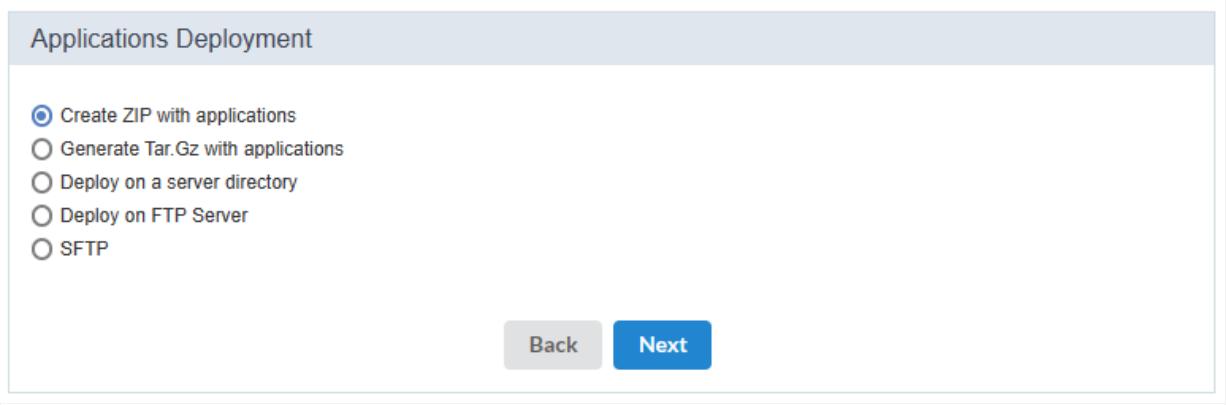
Generated Files

At this stage, the way in which the publication will be carried out will be defined.

- **Generate ZIP with applications** - In this option, at the end of the publication, a button will be displayed to download the .zip file generated with the publication files. We recommend using it to publish in production environments configured on Windows.
- **Generate Tar.Gz with applications** - A .Tar.Gz will be generated with the publication files, at the end of the process the file download will be displayed. This type of compression is recommended for publishing in production environments configured on Linux or MacOS.
- **Publish to directory on the server** - Allows direct publishing to a local directory, where Scriptcase is installed. This type of publication is recommended for servers that share the development and production environment. The publication files will be generated within the indicated folder, therefore, we recommend using an exclusive folder for publication. For example, when indicating the C:\Program Files\NetMake\v9-php81\wwwroot\project_name directory, the application files, and the _lib directory with css and themes will be placed directly in the project_name folder.
- **Publish to FTP server** - Sends the publication files directly to the configured FTP server. The publication files will be placed in the indicated directory, therefore, we recommend using a specific directory for the publication.
- **SFTP** - Just like FTP, publishing via SFTP sends the publication files directly to the configured server, placing the publication files in the specified directory.

Generate ZIP or Tar.Gz with applications

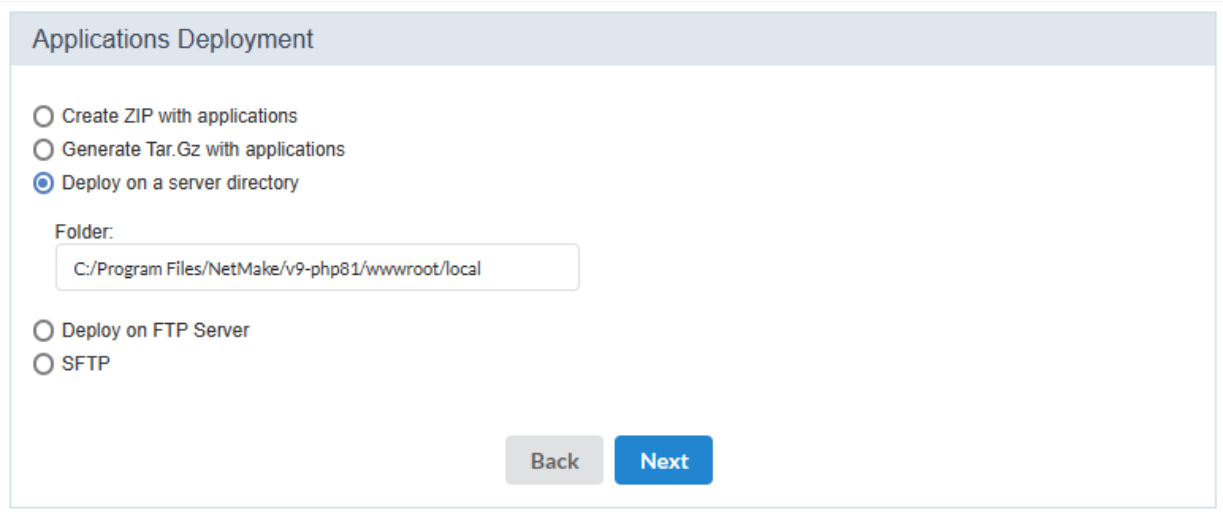
The Scriptcase will generate a file **Zip** or **Tar.Gz**. In the end of this process, will be displayed a report of the deployed applications with the download link for download the project.



The file **ZIP** or **Tar.Gz** generated, must be decompressed only in the deployment server, to avoid that the files be corrupted.

Deploy on a server directory

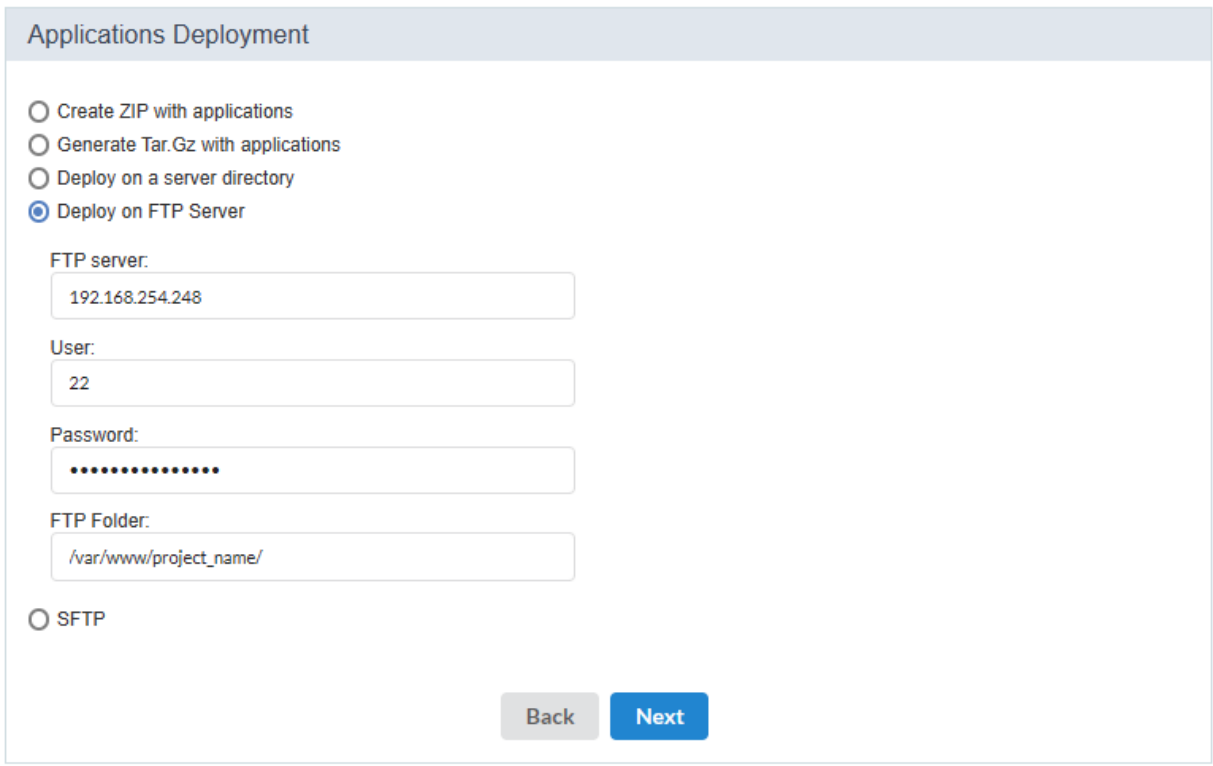
Deploys the project in the informed directory. We must inform the absolute path to the deployment directory.



This option should be used in the case that the deploy is going to be in the same sever where ScriptCase is installed, otherwise this option is not valid.

Deploy on FTP server

This option generates the files directly in the FTP remote server.

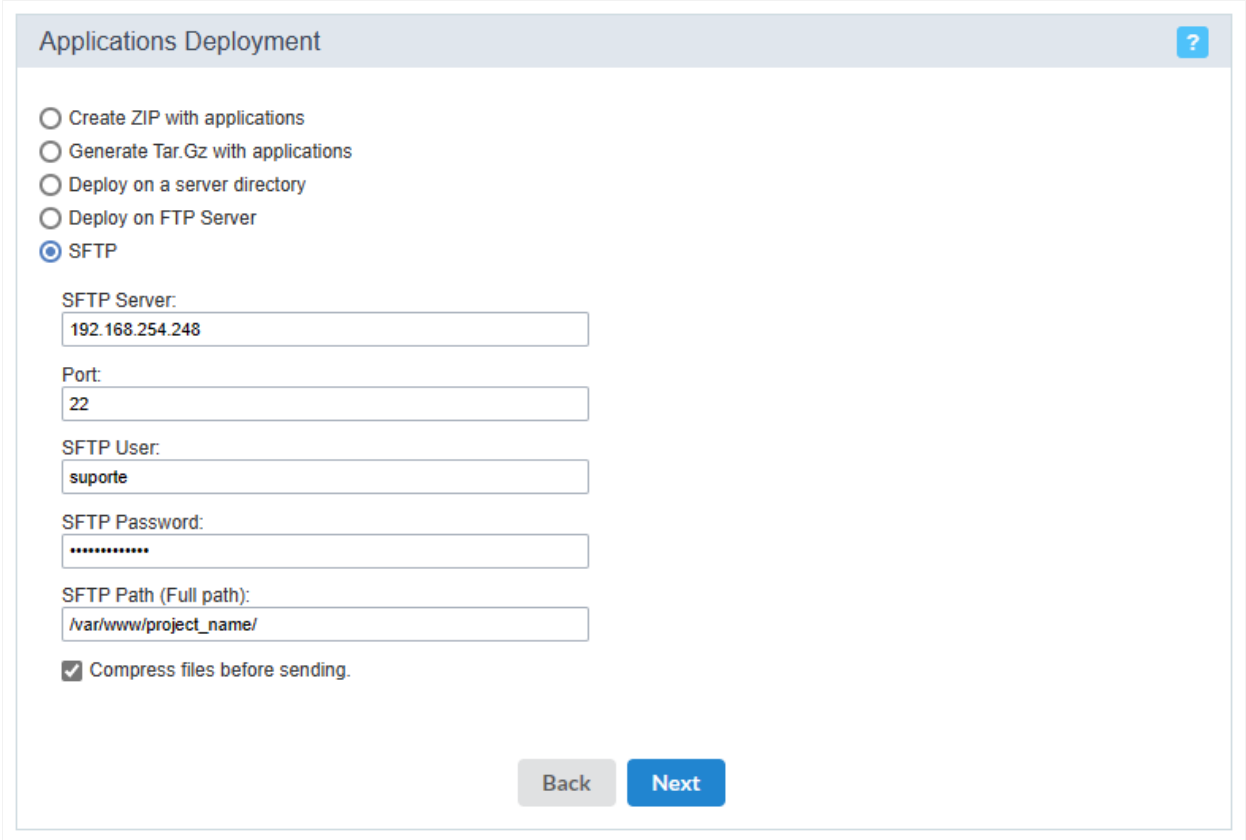


Options	Description
FTP server	Server where you will deploy your project. Inform the domain or the server's IP.
User	User for the FTP server in the production, remembering that the user need to have permission to write in the server folder, otherwise it will not work.
Password	Password for the FTP user.
FTP Folder	Server directory where the project will be deployed in.

To use this option is necessary that you have a server that provides the FTP access and an user with write permissions.

Deploy on a SFTP server

Deploys the application automatically on a remote SFTP server.

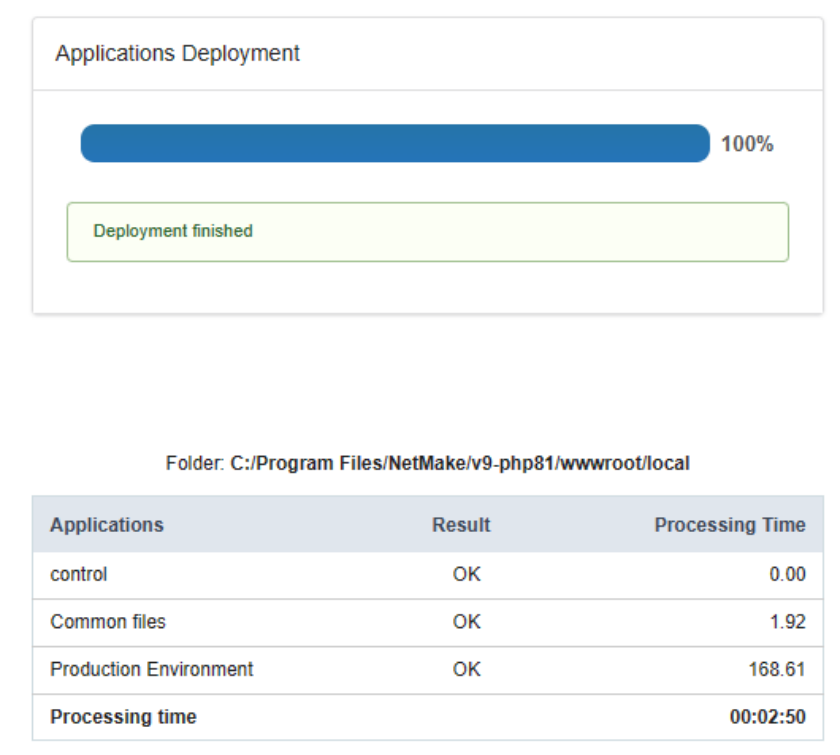


Options	Description
SFTP server	Server where you will deploy your project. Inform the domain or the server's IP.
Port	Defines the port that will be used when connecting to the server.
User	User for the SFTP server in the production, remembering that the user need to have permission to write in the server folder, otherwise it will not work.
Password	Password for the SFTP user.
SFTP Folder	Server directory where the project will be deployed in.
Compress files before sending	This option defines if the files will be compressed before being sent to the server or if they will be sent one by one. We recommend checking the option so that the files are compressed before sending, in this way the publication tends to be much faster, especially when publishing with a large number of files.

To use the **SFTP** option is necessary that the user have permission to write on the server folder.

Deployment log

At the end of the process you will receive a report with all the applications deployed and a download link of the project ready to be published.



Production Environment Overview

The **production environment** gathers the settings essential for the functioning of the published applications. In it, the APIs used in projects and connections must be configured. It is also possible to perform some configurations basic on your web server, such as setting the PHP timezone between others

The production environment can be implemented in two ways.

1. **Shared between two or more projects**: In this case, the settings are centralized in a single environment. The advantage is the ease in maintaining the settings, since all changes will be made in a single place.
2. **Individually configured for each project**: In this case, each project will have its own production environment with specific settings adjusted according to the project's needs.

Both have their advantages and disadvantages and can be adopted depending on the infrastructure and requirements of the projects in question. The important thing is to ensure that the production environment is correctly configured to guarantee the proper functioning of the applications or projects published on the server.

Environment prerequisites

To publish projects and use the production environment, the following prerequisites must be met:

- **Web Server** configured and compatible with PHP versions approved for Scriptcase applications such as:*Apache*, *IIS* or*nginx*.
- **PHP** approved for use in the tool's applications. Check the compatibility list on the Typical deploy and Advanced deploy page.
- **Database extension** used in the project connection, enables the creation of the connection.

Accessing the production environment

Access must be carried out using the browser of your choice, as long as it is compatible. Check the list below.

The default path when performing a typical deploy is: `/project_name/_lib/prod` already in the advanced deploy the directory will be defined by the developer at the time of deploy.

Examples of accessing prod in a typical deploy

- 180.204.163.144:8092/project_name/_lib/prod/
- mysc.domain.net/project_name/_lib/prod/

Examples of accessing prod in an advanced deploy

- 180.204.163.144:8092/production/
- mysc.domain.net/project_name/prod/

Compatible Browsers

Chrome	Any version
Firefox	Version 4 or higher
Edge	Any version
Opera	Version 9 or higher
Safari	Version 5.2 or higher

Configuration Options

The production environment has miscellaneous available configurations, click on the desired option to access the documentation section referring to the resource.

Database Connections

- Create New Connection - Allows creating connections used in the production environment. Make sure the connection name matches what is expected by the application.
- Edit Connection - Allows modifying the data of a previously created connection.
- Rename Connection - Allows changing the name of an existing connection.
- Pending connections - Lists connections for published projects that are using the production environment and have not yet been configured.

Production Configuration

- Configure production environment - Allows adjusting various environment settings such as file lifetime, time zone, and other options.
- Applications directory - Registration screen of directories of published projects, which use the production environment.
- Incompatible applications - Verifies application compatibility with the production environment.
- Configure password recovery - Defines the email to be used for password recovery in the environment.
- Update production environment - Automatically updates production environment files.
- API - Configuration screen for APIs used in the development environment.
- Change Password - Allows changing the password of the production environment.
- **Help** - Link to access this documentation page.
- **Diagnosis** - Accesses the diagnostic file of your environment, where you can view server configurations.
- **Logout** - Performs a secure logout, ending the session in your production environment.

First access

To access the Scriptcase production environment, you must use your preferred web browser.

Accessing prod

The location of the production environment may vary depending on the type of deployment performed.

In a typical deploy, the default access directory is `/project_name/_lib/prod` .

However, if you have performed an advanced deploy, the location of the "prod" directory will be defined by the project developer at the time of deployment.

Examples of accessing "prod" in a typical deployment:

- 180.204.163.144:8092/project_name/_lib/prod
- mysc.domain.net/project_name/_lib/prod

Examples of accessing "prod" in an advanced deployment:

- 180.204.163.144:8092/production
- mysc.domain.net/project_name/prod/

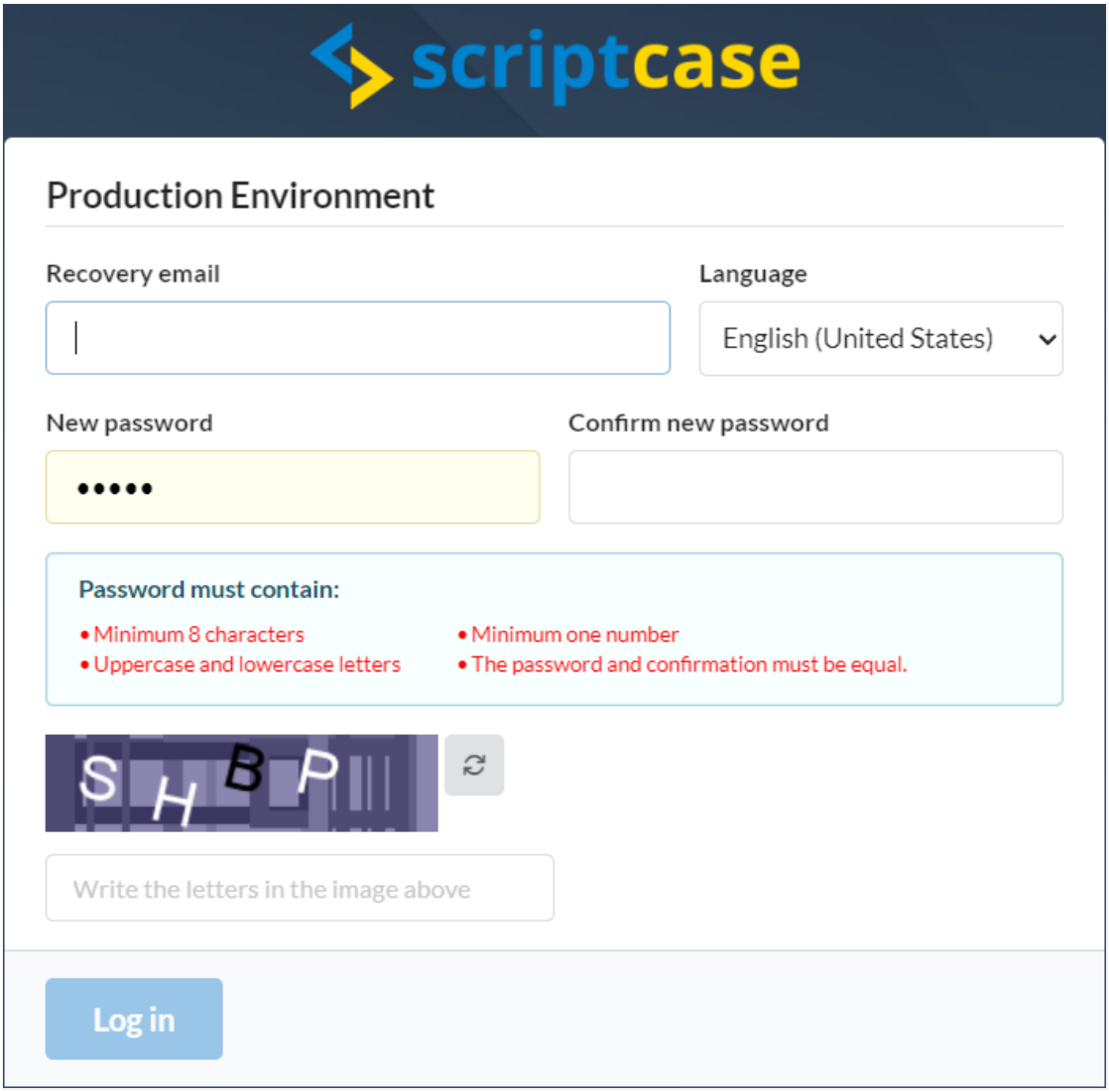
Check the list of supported browsers

Chrome	Any version
Firefox	Version 4 or higher
Edge	Any version
Opera	Version 9 or higher
Safari	Version 5.2 or higher

Setting a new password

On your first access to the production environment, you will need to set the password.

It is essential to emphasize that the production environment stores sensitive information, so it is of utmost importance to set a strong password. We recommend choosing a combination of alphanumeric characters, uppercase and lowercase letters, as well as symbols, to create a strong and secure password.



Email

Enter an email address for password recovery in the production environment. This field is optional and can be changed later within the production environment.

The password recovery process will be exclusively conducted through the provided email.

Language

Set the language in the production environment. Currently available in the following languages: Portuguese (Brazil), English, and Spanish.

New Password

Define a new password.

Confirm new password

Re-enter the password defined in the **New password** field. The two passwords must be identical.

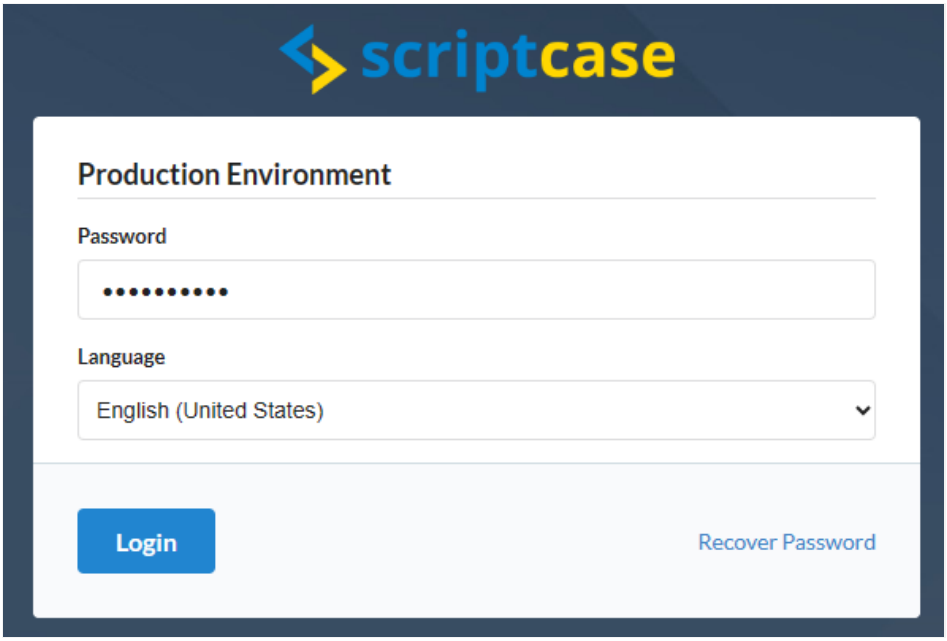
Captcha

Enter the same letters you see in the image.

Accessing with the new password

When accessing the production environment again, use the password defined during the first access.

Remember that the password must be at least 8 characters long and include at least one uppercase letter, one lowercase letter, and one number.



Password

Enter the password defined during the first access to your production environment.

Language

Set the language in the production environment. Currently available in the following languages: Portuguese (Brazil), English, and Spanish.

Password recovery

The password recovery option will be exclusively available through the email registered during the password reset step or in the Configure Password Recovery menu. Learn more about this feature by clicking here.

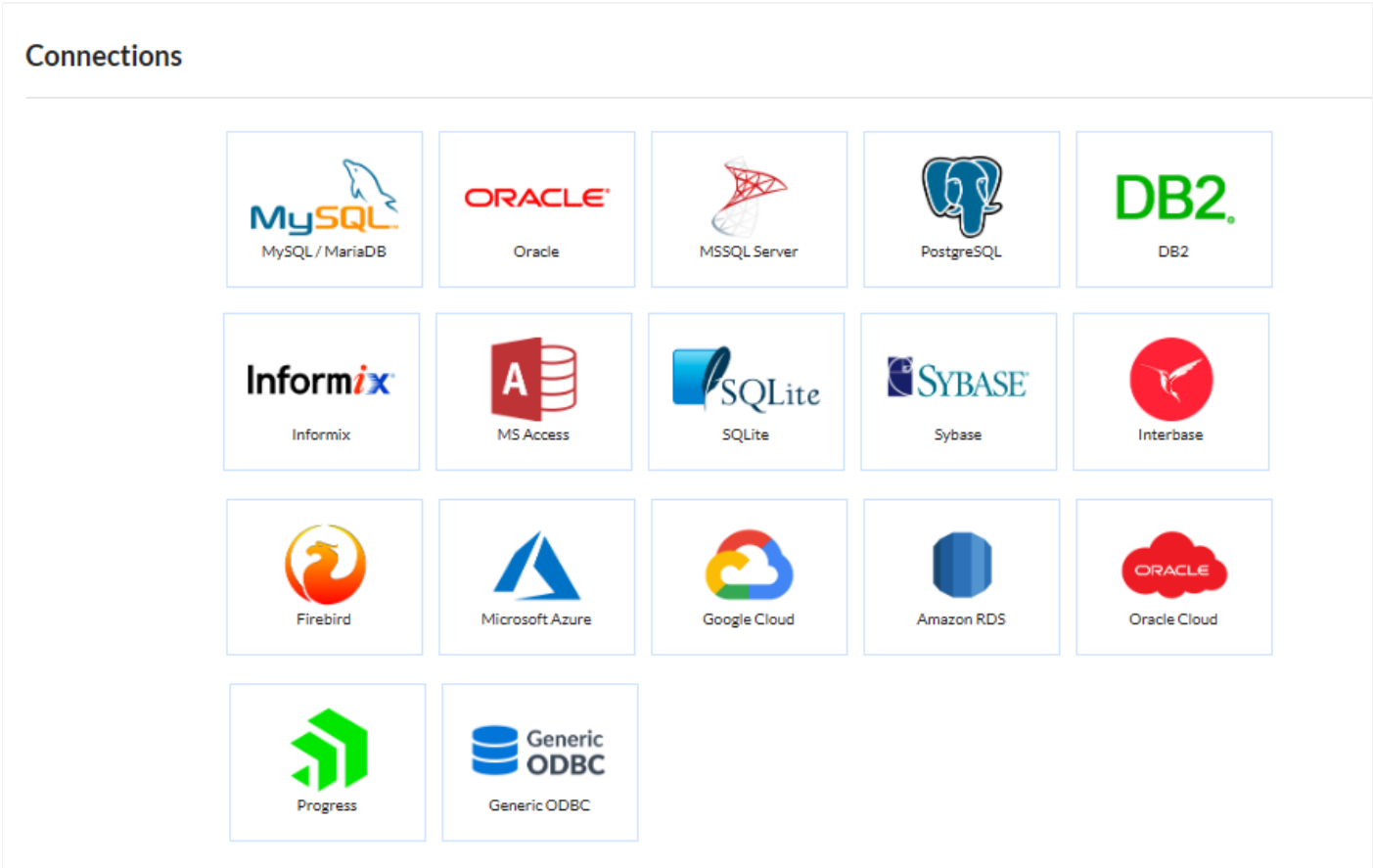
Database Connections

In this section, you will find all the settings related to database connections in the production environment. Here, you can configure and manage the connections that will be used by your projects.

Create new connection

In this section, you can create connections to the databases used by your project.

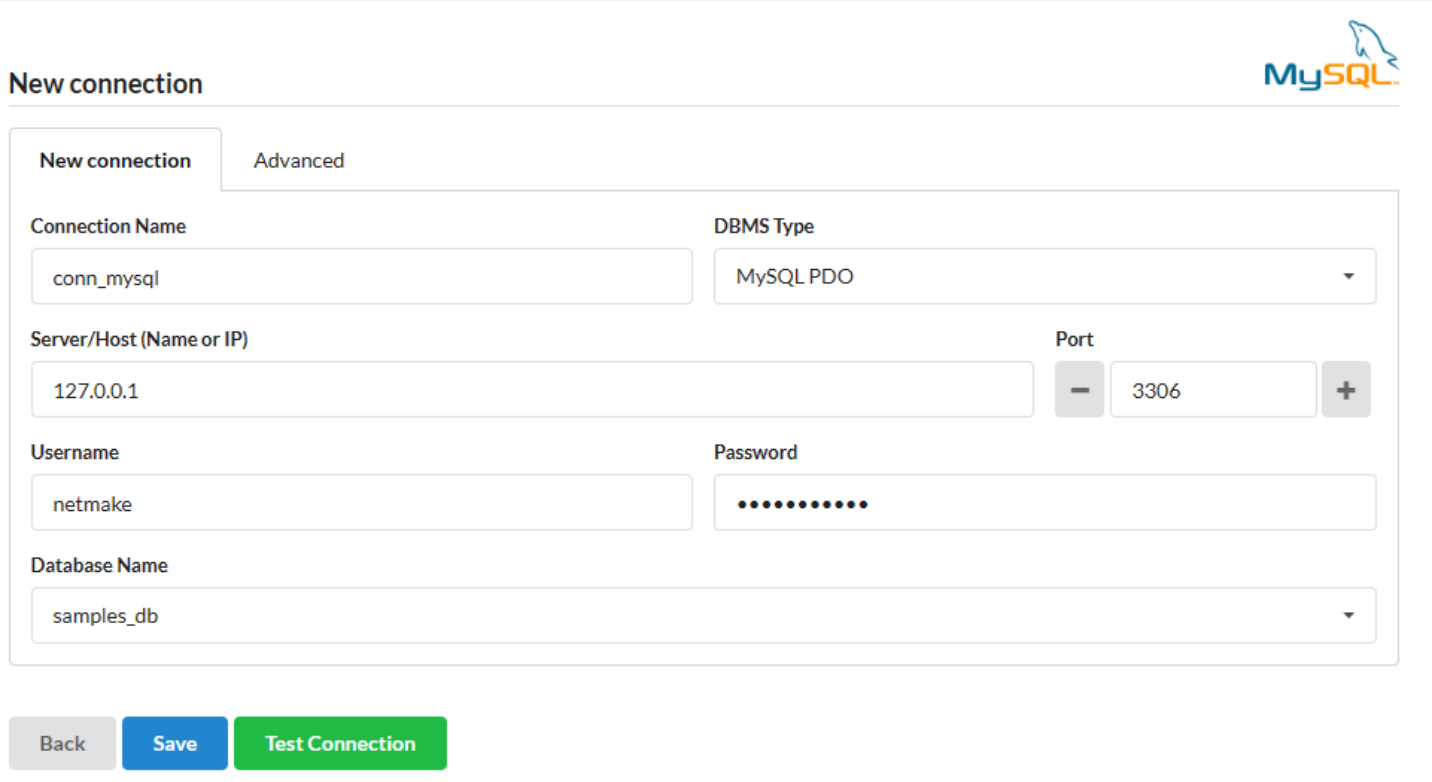
The connection name in the production environment should match the name used during the deployment of the project.



Next, you need to provide the Database Server. If your database is hosted on the same server as the applications, you can use 'localhost' to refer to it. If the database is on a different server, you can use the machine name or IP address. Additionally, provide the login credentials (if applicable) to access the database.

The connection in the production environment should have the same name as the connection used by the project in the development environment.

Example of MySQL connection screen



The establishment of the connection in production is similar to that performed in the development environment. Therefore, if you need more details to configure the connection, access the connection pages and check the configuration details.

- **Connection name**: Specify the name of the connection in the production environment. Make sure to use the name expected by your project, as this information is defined during the deployment.
- **Database management system model**: Select the driver to establish the connection.
- **Database Server**: Allows connecting via the server's domain or IP address.
- **Port (default 3306)**: Select the port to establish the connection.
- **User**: Provide the database user to establish the connection.
- **Password**: Enter the database password to establish the connection.
- **Database**: Specify the database to be used in the connection.

Edit existing connection

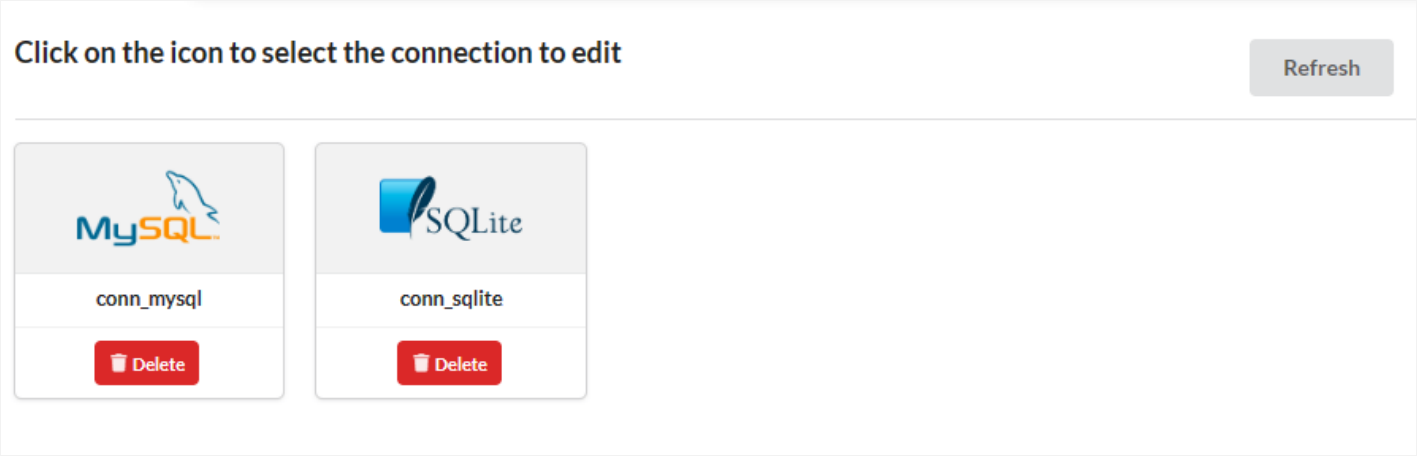
All existing connections in the production environment will be listed here. To edit a connection, simply click on the desired connection card, and the connection configuration screen with the current data will open.

In this screen, you can also delete existing connections in the environment by clicking the **Delete** button.

Before modifying or deleting a connection, make sure it is not in use.

The refresh button reloads the list of connections.

Example of connection list

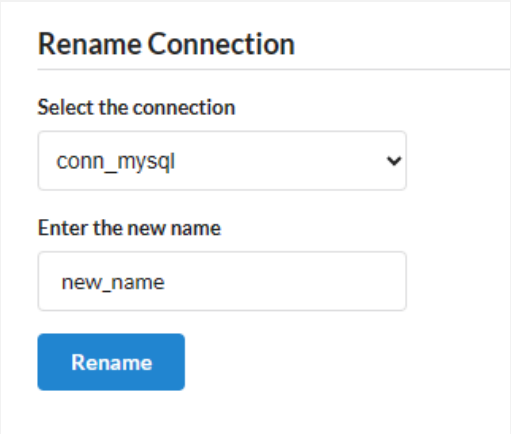


The establishment of the connection in production is similar to that performed in the development environment. Therefore, if you need more details to configure the connection, access the connection pages and check the configuration details.

Rename connection

This option allows you to rename a connection in the production environment.

In the **Select the connection** field, all existing connections will be listed. Select one of them and enter the desired new name.

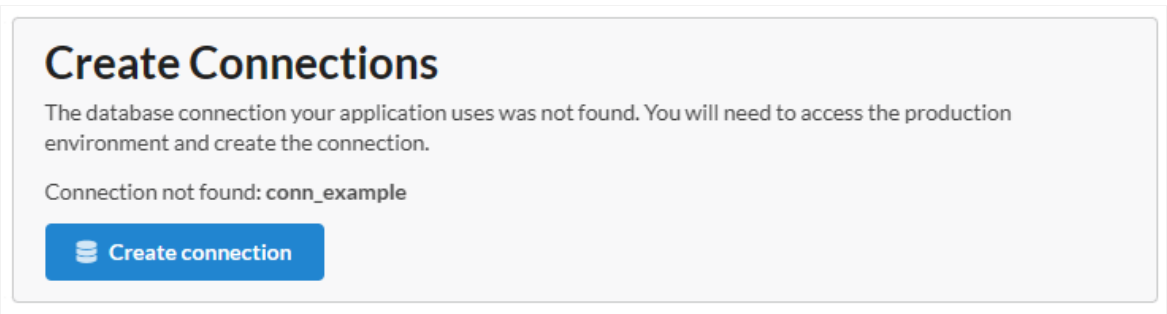


This procedure **does not change the name of the connection used by the project**. The connection in the production environment should have the same name as the connection used by the project in the development environment.

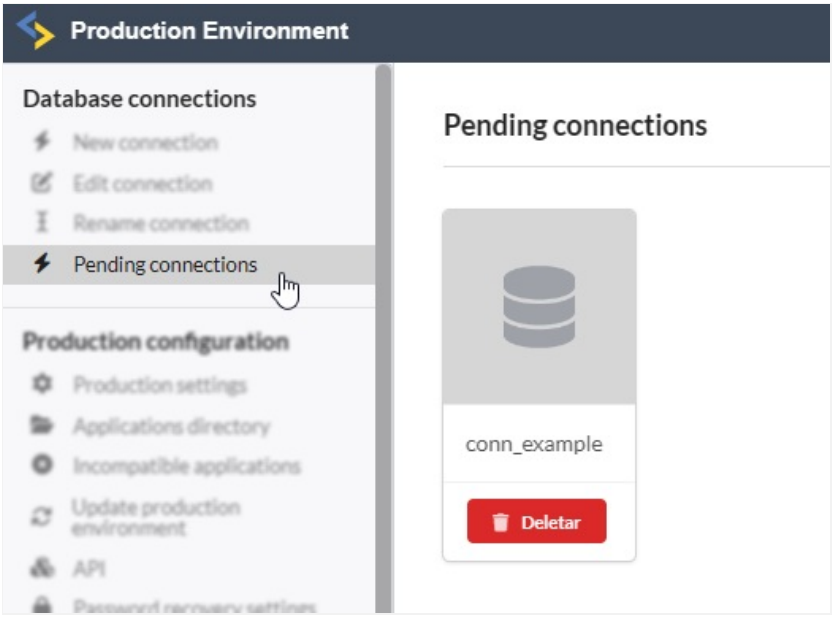
Pending connections

Displays the list of pending connections, which have not yet been created, for projects using the production environment.

When accessing a published project, which has not yet had its connection configured, the connection not found screen will be displayed.



When accessing the production environment used by the project, access the item Pending connections. On this screen, all connections, from projects that use this production environment, that have not yet been configured will be listed.

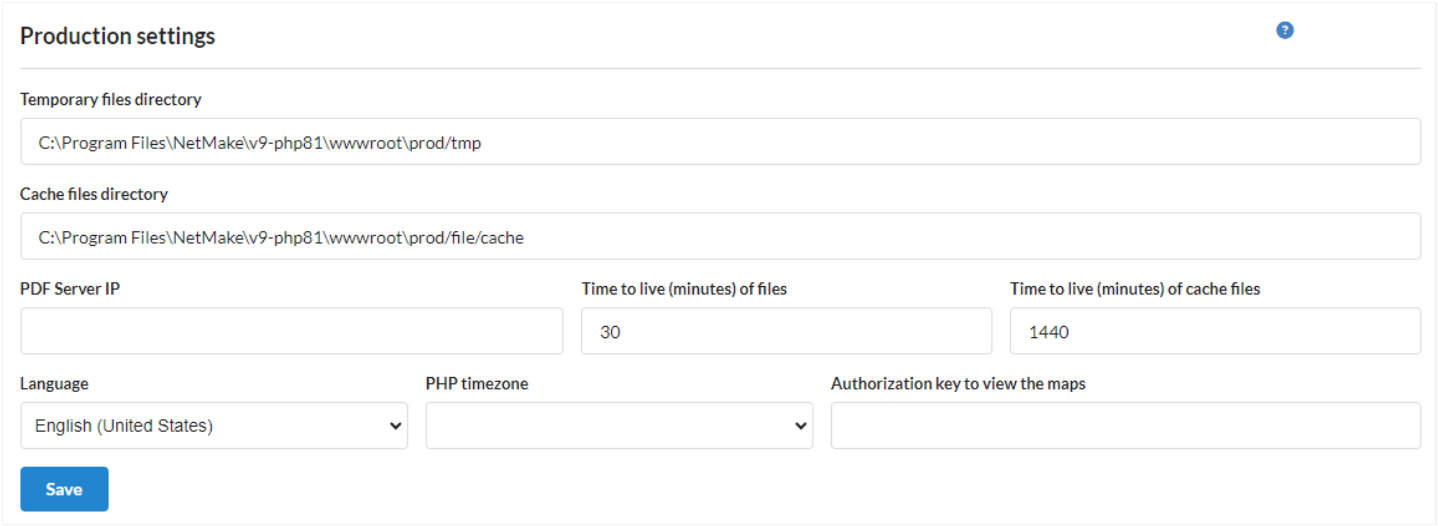


Production Configuration

In this section, you will find the settings related to the production environment, where you need to define some essential resources for the functioning of the deployed project.

Production Settings

Configure the available settings in the production environment below.



Temporary Files Directory

This directory is used to temporarily store files generated by exports and upload fields in the project.

Cache Files Directory

In this directory, cache files, commonly used images, and text files are stored for applications.

PDF Server IP

This field defines the IP address of the server where application PDF files will be generated.

This configuration should only be used to resolve any PDF generation blocking issues related to the server's IP or hostname. If there are no problems with your PDF generation, leave this field blank.

Cache Files Lifetime (Minutes)

Here, you can define the time, in minutes, that files in the Cache Files Directory should persist before being refreshed or deleted.

Files Lifetime (Minutes)

This field determines the time, in minutes, that files in the Temporary Files Directory should persist before being removed.

Language

Sets the language used in the production environment.

PHP Timezone

Specify the timezone that will be used by your project. If you leave this field blank, the PHP environment's timezone will be used.

Authorization Key for Map Viewing

To use Google Maps features in the environment, enter the authorization key for the Google Maps API here.

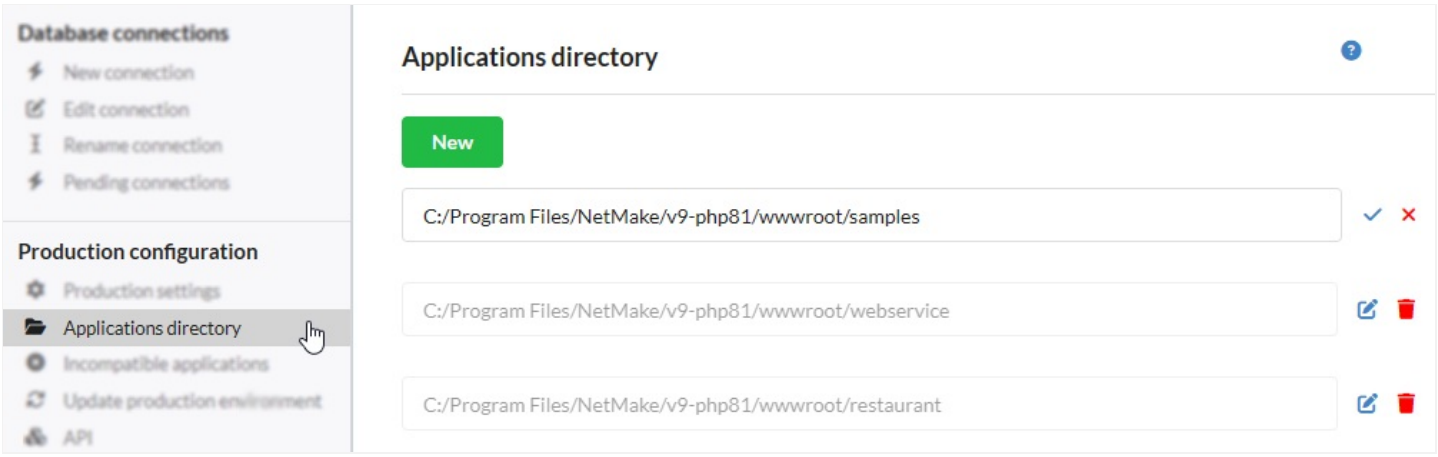
Learn how to create the key by accessing the article Google Maps API Credentials

Applications directory

On this screen, the developer must register the directories that have applications published and linked to the production environment in configuration.

This registration is necessary for these directories to be listed in the item Aplicações desatualizadas in order to verify the compatibility of the applications with the production environment being configured.

The directory must be registered informing the complete path from the root of the server.



Click **New** to add more directories and without **Save** to store the changed directories.

It is possible to register directories that do not yet have applications, but that will be used in a future publication.

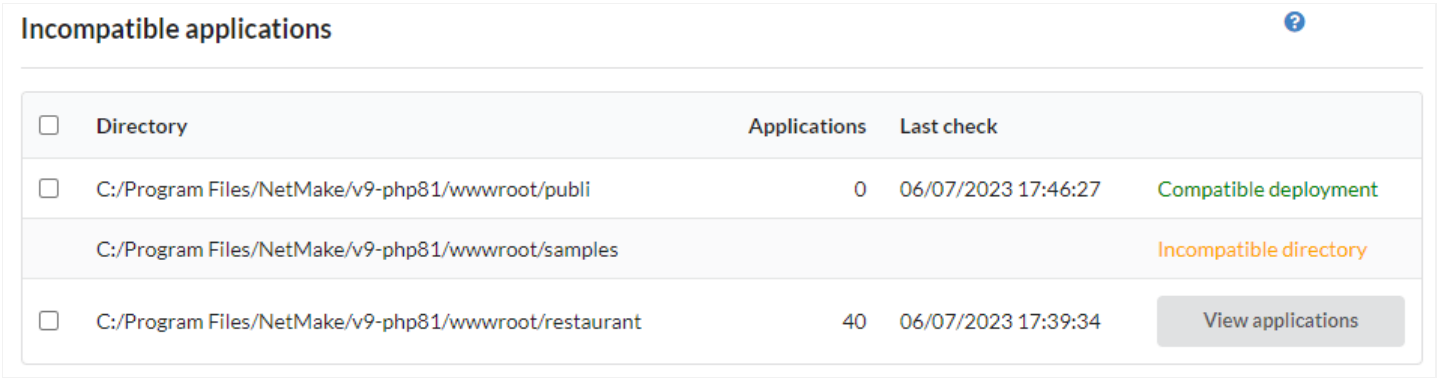
Incompatible applications

On this screen, all the directories registered in the **Directory of applications** section will be listed so that the developer can verify the compatibility of the applications published in these directories with the current production environment.

This verification will be carried out by comparing the **current version of the production environment** with the **version of the production environment used by Scriptcase at the time of publication** of these applications.

Applications will be considered incompatible if they have different versions between the production environments mentioned above.

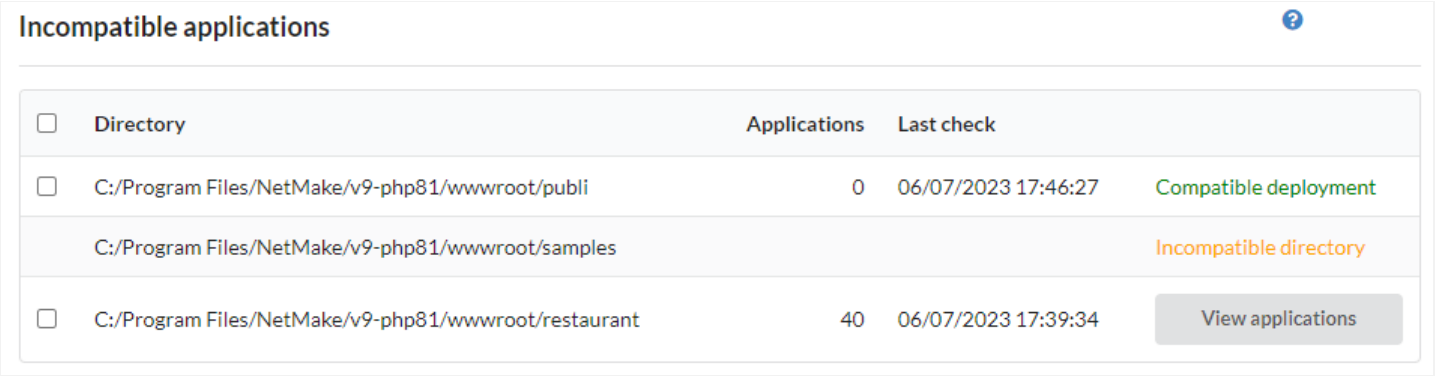
For example, if Scriptcase's production environment is in version 2.0.000 and the server's production environment is in version 1.0.000, the applications published by this Scriptcase in this production environment will be considered incompatible.



Check application compatibility

To carry out the verification, select the desired directories and click on the **Validate** button. When finished, a confirmation message will be displayed.

Then the scan results will be displayed, as in the image below.



Directory

Lists the directories registered in the **Application Directory** section.

Applications

Total directory applications that are incompatible with the environment the production environment in configuration

Last Check

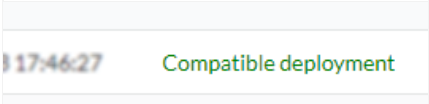
Date and time of the last verification performed on the directory.

Action column

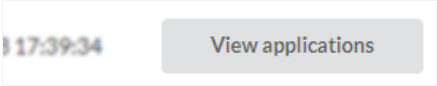
This column will return the result of the check.

- **Compatible deployment message** - All applications in the directory are compatible with the current production environment.
- **View applications button** - One or more applications in the directory are incompatible with the current production environment, click the button to list the applications.
- **Incompatible directory message** - The specified directory has no applications.

Example directory with compatible applications



Example directory with incompatible applications



See applications

When clicking on the button, all incompatible applications in the directory will be listed and the details.

Application

In this column, the names of the applications that are incompatible with the current production environment will be displayed.

Publication version

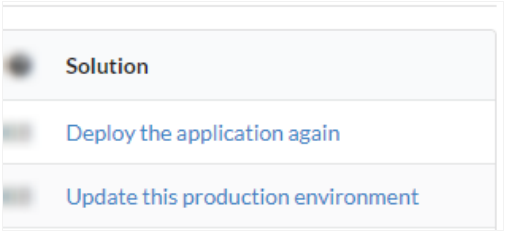
Informs the version of the production environment, used in the development environment, by Scriptcase at the time the application was published.

Version of this environment

Reports the current version of the production environment being configured.

Solution

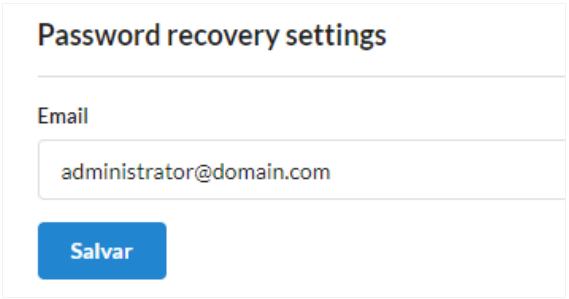
This column displays the action that must be taken to make the applications compatible with the production environment and a link to a tutorial regarding the informed solution..



Configure Password Recovery

In this section, you define the email address that will be used for password recovery in the environment.

In this process, a temporary code will be sent to the email address set in this configuration. Therefore, it is recommended to use an email with restricted access.



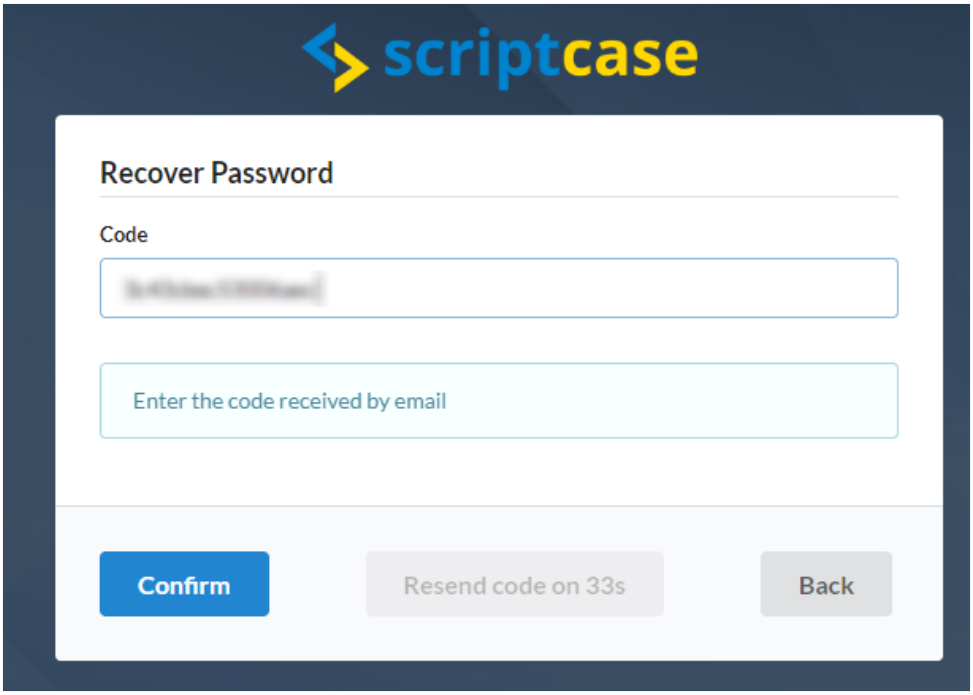
Follow the steps below to recover the password.

Recover Password in the production environment

When clicking the **Recover Password** link, a temporary code will be sent to the provided email address. This code needs to be entered on the password recovery screen, which will be displayed after sending the email.

Example of the code information screen

Enter the code in the field and click **confirm**.

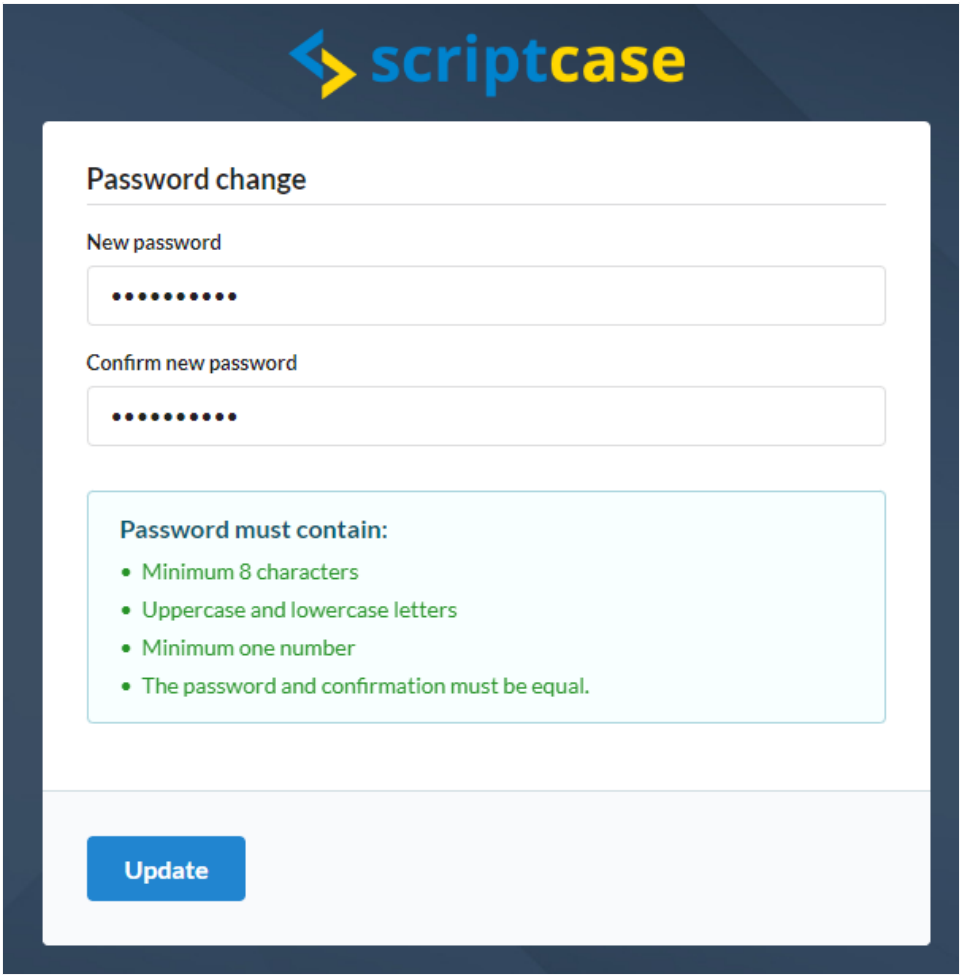


Password Reset

The password reset screen will appear. Enter the new password, following the specified criteria, and click Change. After confirming the change, the login screen will be displayed.

It is essential to emphasize that the production environment stores sensitive information, so it is of utmost importance to set a strong password. We recommend choosing a combination of alphanumeric characters, uppercase and lowercase letters, as well as symbols, to create a strong and secure password.

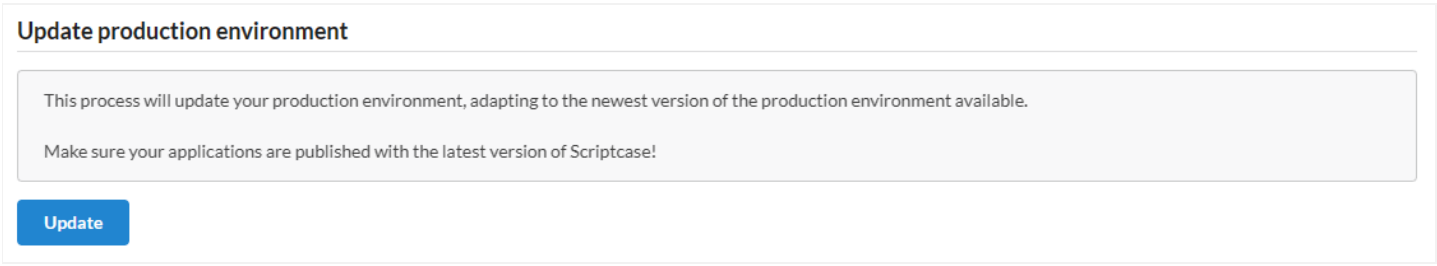
Example of the new password definition screen



Update Production Environment

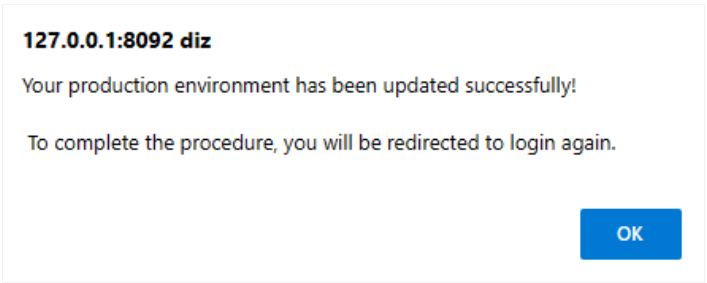
This feature allows you to quickly and securely update the production environment automatically.

Automatic update screen



After completing the update, a confirmation screen will be displayed.

Update confirmation screen



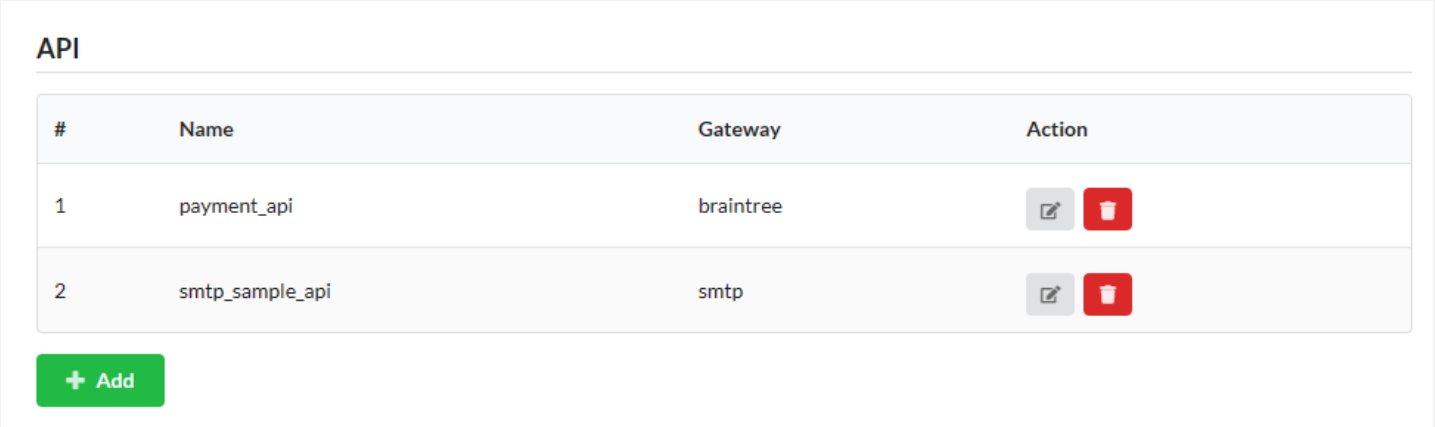
API

When publishing the project, the **API data configured in the development environment** within Scriptcase **is not sent**.

Therefore, it is necessary to configure the API in the production environment during the deployment. Remember that the API must have the same name used in the development environment so that applications can use them.

The API configuration in the production environment is similar to the one in the development environment. If you have any doubts, check the Scriptcase API documentation.

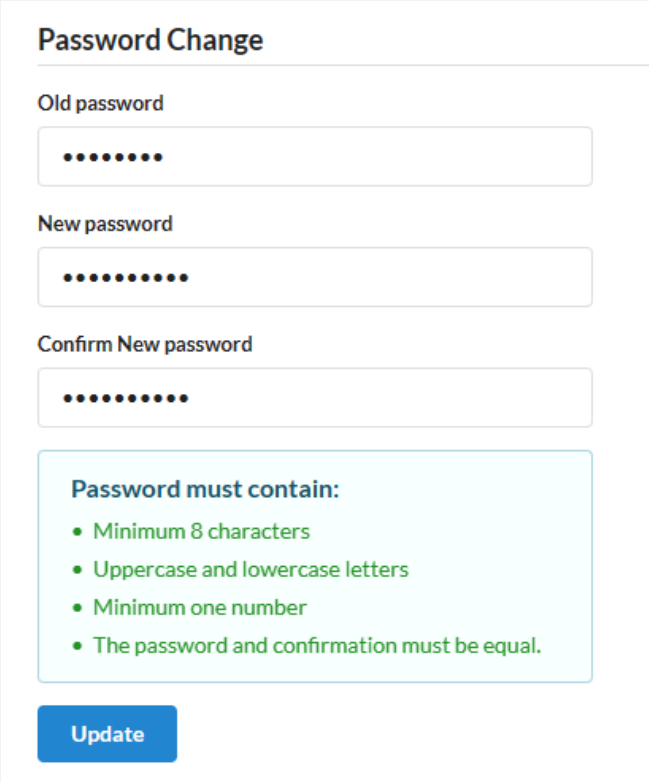
Example of the API list in the production environment



Change Password

Allows you to change the current password of the production environment. You need to provide the current password to validate the change.

Password reset screen



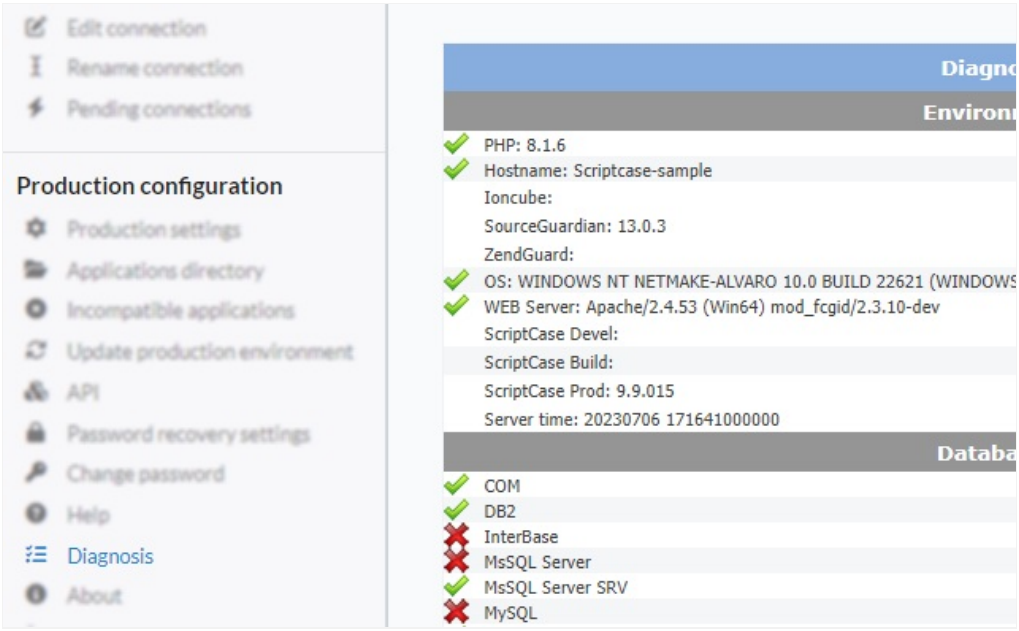
Help

Link to the documentation page on the website regarding the production environment.

Diagnosis

Displays your Web server configuration information, such as: PHP version used, enabled drivers, among other important information.

It is only possible to view this information after logging in to the production environment.



About

Displays information about the production environment.

Logout

Logs out of your production environment, ending the current session.

Scriptcase Macros Overview

What are macros?

Macros are functions created by Scriptcase developers to help you customize your application and coding. Using Macros simplifies your code editing as well as your validation processes.

The code editor within Scriptcase provides you some functionalities which allow you to change the code editor to wide screen or to display the sidebars menus with some options.

Once you're inside the event code editor, you will see the right side bar menu with the available macros being displayed with a green point

Scriptcase's macros always star you can type with a "sc_". A hint is that "sc_" and hit **ctrl + spacebar** in your keyboard to check a list of available macros in the event you are.

Where can they be used?

You can use the macros within the application events, depending on the scope described in the article Events X Macros and within internal or external libraries from Scriptcase. application methods and fields lookups.

SCRIPTCASE MACROS

Scriptcase has macros and special variables that allow the user to manipulate events, buttons, applications, security control, manipulate with dates, etc.

Follows a table of all macros and special variables from **ScriptCase**, including the list of applications that works for each one.

MACROS AND SPECIAL VARIABLES	
SQL	
sc_begin_trans ("Connection")	This macro starts a set of transactions in the database.
sc_change_connection ("Old_Connection", "New_Connection")	This macro dynamically change the application connections.
sc_commit_trans ("Connection")	Makes all data modifications performed since the beginning of the transaction a permanent part of the database.
sc_concat ()	This macro is used to combine two or more strings and/or fields from the table.
sc_connection_edit ("Connetion_Name", $arr_conn)	This macro edits an existing connection at runtime.
sc_connection_new ("Connection_Name", $arr_conn)	This macro creates new connections dynamically.
sc_error_continue ("Event")	This macro deactivates the Scriptcase standard database error treatment message for an event.
sc_error_delete	This macro configure the variable that contains the database error message that can occurs during the exclusion of a record.
sc_error_insert	This macro configure the variable that contains the database error message that can occurs during the addition of a record.
sc_error_update	This macro configure the variable that contains the database error message that can occurs during the update of a record.
sc_exec_sql ("SQL Command", "Connection")	This macro execute SQL commands passed as parameter or a SQL command in the SQL field action type.
sc_lookup (Dataset, "SQL Command", "Connection")	This macro executes a SELECT command stored in the second parameter and returns the data in a variable.
sc_lookup_field (Dataset, "SQL Command", "connection_name")	This macro executes a SELECT command stored in the second parameter and returns the data in a variable.
sc_reset_change_connection	This macro erases the changes made using "sc_change_connection".
sc_reset_connection_edit	This macro undoes the connection edits made by macro "sc_connection_edit".
sc_reset_connection_new	This macro undoes the connections made by the macro "sc_connection_new".
sc_rollback_trans ("Connection")	This macro discards a set of transations in the data base.
sc_select (dataset, "SQL Command", "Connection")	This macro executes the commands passed in the second parameter and returns the dataset in a variable.
sc_select_field ({Field})	This macro modify dynamically a field that will be recovered in the grid.
sc_select_order ("Field")	This macro modifies the grids "ORDER BY" clause field dynamically.
sc_select_where (add)	This macro dynamically adds a condition to the grid WHERE clause.
sc_set_fetchmode (parm);	This macro allows to change the type of return from the dataset of the select commands
sc_sql_injection ({My_Field}) or ($My_Variable)	This macro is used to protect the field/variable against "SQL injection" attempts.
sc_sql_protect (Value, "Type", "Connection")	This macro protects the value passed as parameter according with the used database.
sc_where_current	This macro is used to make a reference of the where clause currently used.
sc_where_orig	Retrieve where clause used in original select command of the application.
Variables - Database Database Variables	Global variables containing database access values used by main connection.
Date	
sc_date (Date, "Format", "Operator", D, M, Y)	This macro calculates and returns increments and decrements using dates.
sc_date_conv ({Field_Date}, "Input_Format", "Output_Format")	This macro converts the date field passed as parameter with an input format to another field with an output format.
sc_date_dif ({Date1}, "Format Date1", {Date2}, "Format Date2")	This macro calculates the difference between two dates (passed as parameters) returning the result in days.
sc_date_dif_2 ({Date1}, "Format Date1", {Date2}, "Format Date2", Option)	This macro calculates the difference between two dates returning the amount of days, months and years.
sc_date_empty ({Field_Date})	This macro checks if a date field its empty returning a boolean.
sc_time_diff ({datetime_01}, "Date_01 Format", {datetime_02}, "Date_02 Format")	Calculate difference in hours, returning the amount of hours, minutes and seconds.
Control	
sc_actionbar_clicked_state ()	Returns the current state of the AJAX button created on the action bar when it is clicked.
sc_actionbar_disable ("button_name")	Disables a button that was created on the action bar.
sc_actionbar_enable ("boton_nombre")	Enables buttons that were created on the action bar.
sc_actionbar_hide ("button_name")	Hides buttons that were created on the action bar.
sc_actionbar_show ("button_name")	Displays buttons that were created on the action bar.
sc_actionbar_state ("button_name", "state_name")	Changes the current state of a button on the action bar.
sc_ajax_javascript ('JavascriptMethodName', array("parameter"))	Executes JavaScript methods within Form or Control events.
sc_alert ("Message", $array)	Displays alert messages with SweetAlert or SweetAlert2, including support for customizable options and toast mode.
sc_api_download (profile, settings, file, destination)	Downloads files using the Storage APIs.
sc_api_gc_get_obj ($app_name , $json_oauth, $auth_code)	Generates the token_code for Google API authentication.
sc_api_gc_get_url ($app_name, $json_oauth)	Generates a URL for Google account authentication in API configuration.
sc_api_storage_delete (profile, file, parents)	Deletes files stored in cloud storage services.
sc_api_upload (profile, settings, file, parents)	Uploads files using the Storage APIs.
sc_apl_conf ("Application", "Property", "Value")	Modifies the execution behavior of another application dynamically.
sc_apl_default ("application", "type");	Defines the behavior of the initial application when the session is lost.
sc_calc_dv (Digit, Rest, Value, Module, Weights, Type)	This macro calculate verifier digits. (checksums)
sc_call_api ($profile, $arr_settings)	This macro allows us to use as integrated Scriptcase APIs
sc_captcha_display ("on/off")	Dynamically controls the display of captcha in the application.
sc_changed ({Field_Name})	This macro returns "true" if the field name have been changed.
sc_confirm ("Message")	This macro shows a Javascript confirmation screen.
sc_decode ({My_Field})	This macro returns the encrypted field or variable to its original value.
sc_encode ({My_Field})	This macro returns the field or variable with the content encrypted.
sc_error_exit (URL / My_Application, "Target");	Interrupts the method being processed and must be used in conjunction with the "sc_error_message" macro
sc_error_message ("Text")	This macro generates an error message.

sc_exit (Option)	This macro forces the application to exit.
sc_field_no_validate ('field_name')	It ignores the validations defined in the field configuration screen, such as: Required field, character types, among others.
sc_getfield ("myField")	This macro assigns the properties of a field to a javascript variable.
sc_get_language	This macro returns the abbreviation of the language being used.
sc_get_regional	This macro returns the abbreviation of the regional settings being used.
sc_get_theme	This macro returns the application theme name.
sc_get_wizard_step	Retrieves the id of the current page, in the transition between the steps of a wizard form.
sc_groupby_label ("My_Field")	This macro dynamically modify the field label displayed in group by lines.
sc_image (Image01.jpg)	This macro loads images passed as parameter to use in the application.
sc_include ("File", "Source")	This macro is used to "include" PHP routines.
sc_include_lib ("Lib1", "Lib2", ...)	This macro is used to dynamically select the application libraries.
sc_include_library ("Target", "Library Name", "File", "include_once", "Require")	This macro includes a PHP file from a library in the application.
sc_label ("field_name")	This macro its used to modify dynamically the grid form field label.
sc_language	This macro returns the application language
sc_link (Column, Application, Parameters, "Hint", "Target", Height, Width)	This macro dynamically creates or modifies links between grid applications and other applications.
sc_log_add ("action", "description")	This macro will add a record into the log table.
sc_log_split ({description})	This macro returns what was inserted in the "description" field of the log table in an array format.
sc_mail_send (SMTP, Usr, Pw, From, To, Subject, Message, Mens_Type, Copies, Copies_Type, Port, Connection_Type, Attachment, SSL)	This macro is used to send e-mails.
sc_make_link (Application, Parameters)	This macro is used to create a string with the link data to another application.
sc_master_value ("Object", "Value")	This macro update a Master Application object from a Detail Application.
sc_redir ('app_name/url', parameter01; parameter02, 'target', 'error', 'modal_height', 'modal_width')	This macro is used to redirect to another application or URL.
sc_reset_apl_default	This macro can reset sc_apl_default macro settings.
sc_reset_global ([Global_Variable1], [Global_Variable2] ...)	This macro deletes session variables received as a parameter or multiple parameters.
sc_send_mail_api ($arr_settings)	Enables dynamic sending of embedded emails with Mandrill and Amazon SES
sc_send_notification ('title', 'message', 'destiny_type', 'to', 'from', 'link', 'dtexpire', 'profile')	Sends notifications dynamically to the system users.
sc_send_sms ($arr_settings)	This macro is used to send text SMS messages
sc_seq_register	This macro provide the register sequential number.
sc_set_global ($variable_01) or ({My_Field})	This macro is used to register session variables.
sc_set_groupby_rule	This macro is used to select a specific GROUP BY rule.
sc_set_language ('String Language')	This macro allows us to dynamically change the application language.
sc_set_regional ('String Regional')	This macros allows to dynamically change the application regional settings.
sc_set_theme ('String Theme')	This macro is used to dynamically define the application themes.
sc_site_ssl	This macro verifies if its been used on a safe/secure site. (https protocol)
sc_statistic (arr_val, tp_var)	Calculates and returns an array with statistical values, from an array with numeric values
sc_trunc_num ({My_Field}, Decimal_Number)	This macro its used to set the number of decimals.
sc_url_exit (URL)	This macro modifies the application exit URL.
sc_url_library ("Target", "Library Name", "File")	This macro returns the path of a file, inside a library, to be used on the applications.
sc_warning 'on' or 'off'	This macro dynamically activates or deactivates warning messages control.
sc_webservice ("Method", "URL", "Port", "Send Method", "Parameters Array", "Setting's Array", "Timeout", "Return")	This macro is used to communicate with a web service.
sc_zip_file ("File", "Zip")	This macro is used to generate ZIP files from a file list and/or directories.
Variables - Totalling Totalling Variables	Variables that contains all the totals (general and for grouping).
Variables - Totalling (group by) Totalling Variables (group by)	Breaking totals variables.
Filter	
sc_where_filter	This macro its used to save the where clause content generated through the filter form.
Security	
sc_apl_status ("Application", "Status")	Controls access by enabling or disabling applications dynamically based on user permissions.
sc_ldap_groups	Retrieves the existing groups in Active Directory (AD).
sc_ldap_login ($server, $version, $user, $password, $dn, $group, $port, $library)	Main macro for LDAP authentication, responsible for establishing the connection with the server.
sc_ldap_logout ()	This macro is used to release the connection after using the macro sc_ldap_login
sc_ldap_search ($filter = 'all', $attributes = array())	This macro is used to perform searches in the LDAP.
sc_ldap_users ($filter = 'all', $attributes = array())	Retrieves LDAP users and their attributes based on the authenticated user's permissions.
sc_reset_apl_conf ("Application", "Property")	This macro deletes all the modifications effected by "sc_apl_conf" macro.
sc_reset_apl_status	This macro deletes all the application security status variables.
sc_reset_menu_delete	This macro restores a menu item structure. (removed by the macro "sc_menu_delete").
sc_reset_menu_disable	This macro its used to enable a menu item structure. (disabled by the macro "sc_menu_disable").
sc_user_logout ('variable_name', 'variable_content', 'apl_redir.php', 'target')	Macro used to log the user out to the system.
Shows	
sc_ajax_message ("Message", "Title", "Parameters", "Parameters_Redir", "String_toast");	Displays customizable messages in Form or Control AJAX events.
sc_ajax_refresh	Dynamically reloads data in Grid applications.
sc_block_display (Block_Name, on/off)	This macro dynamically show/hide the fields of a specific block.
sc_change_css ("attribute", "value", "field_name")	Allows manipulating CSS properties of query fields and rows.
sc_event_hint ('field_name', 'help message', maximum_width)	Allows adding a help text to links created from an onClick Ajax event.
sc_field_color ("Field", "Color")	This macro changes the color of a determined field text.
sc_field_disabled ("Field_Name = True/False", "Parameter")	This macro is used to block a field to get any data that would be typed on it.
sc_field_disabled_record ("Field_Name = True/False", "Parametre")	This macro has the objective to block the typing on determined fields in the Forms.
sc_field_display ({My_Field}, on/off)	This macro will display or hide a specific field dynamically.
sc_field_init_off (Field1, Field2,...)	This macro is intended to inhibit the query fields on the initial load.
sc_field_readonly ({Field}, on/off)	This macro dynamically sets a form field attribute to '"Read-Only"
sc_field_style ({My_Field}, "Background-Color", "Size", "Color", "Family", "Weight")	The sc_field_style macro dynamically customizes the CSS styling of Grid fields in PHP applications created with Scriptcase.
sc_foot_hide ()	This macro disables the footer display.

sc_format_num ({My_Field}, "Group_Symb", "Dec_Symb", "Amount_Dec", "Fill_Zeros", "Side_Neg", "Currency_Symb", "Side_Currency_Symb")	This macro its used to format numerical values.
sc_format_num_region ({My_Field}, "Qtde_Dec", "Insert_Zeros", "Monetary_Sym")	This macro has the objective to format numbers, using the regional settings
sc_form_show 'on' or 'off'	This macro dynamically show or hide the form.
sc_get_groupby_rule ()	This macro provides the name of the Group By rule running at the time.
sc_head_hide ()	Inhibits the header display.
sc_hide_groupby_rule ('group1', 'grop2', 'group3')	Macro used to disable Group By rules.
sc_set_focus ('Field')	This macro its used to set the focus to a specific field in a form application.
sc_text_style ({My_Field}, "Background-Color", "Size", "Color", "Family", "Weight")	This macro allows to dynamically modify the text style of a grid field.
sc_widget_config ($arrayOptions)	Dynamically changes the visual properties of a widget.
sc_widget_data ('dataName')	Returns comparative data from an index widget, such as value, period, and variation.
sc_widget_name	Returns the name of the widget being executed.
Buttons	
sc_btn_copy	This macro returns "true" when the "copy" button is selected in a form.
sc_btn_delete	This macro returns "true" when the "Delete" button is selected in a form.
sc_btn_disabled ("name_button", "status")	It is intended to dynamically enable or disable a toolbar button.
sc_btn_display ("Button_Name","on/off")	This macro shows and hides buttons on the toolbar in execution time.
sc_btn_insert	This macro returns "true" when the "Add" button is selected in a form.
sc_btn_label ("btn_name", "new_label")	This macro is used to change the label of the buttons.
sc_btn_new	This macro returns "true" when the "Add New" button is selected in a form.
sc_btn_update	This macro returns "true" when the "Save" button is selected in a form.
PDF	
sc_set_export_name ("export_type", "file_name")	Changes the name of files exported by the query
Variables - Authentication Authentication Variables	User/Password for the WEB server.
Menu	
sc_appmenu_add_item ("Menu_Name", "Id_Item", "Id_Parent", "Label", "Aplication", "Parameters", "Icon", "Hint", "Target","mega menu")	Adds an item to a dynamic menu previously created using the sc_appmenu_create macro.
sc_appmenu_create ("Menu_Name")	Initializes a dynamic menu structure to allow adding items via code.
sc_appmenu_exist_item ("Menu_Name", "Id_Item")	Checks if a menu item exists in a dynamic menu and returns true or false.
sc_appmenu_remove_item ("Menu_Name", "Id_Item")	Removes a menu item dynamically from a menu created with sc_appmenu_create.
sc_appmenu_reset ("Menu_Name")	Resets the internal array used in the dynamic creation of a menu application.
sc_appmenu_update_item ("Menu_Name", "Id_Item", "Id_Parent", "Label", "Aplication", "Parameters", "Icon", "Hint", "Target")	This macro updates a menu item.
sc_btn_disable ('button_id', 'on/off')	This macro is used to disable Menu buttons.
sc_menu_delete (Id_Item1)	This macro remove items of the menu structure.
sc_menu_disable (Id_Item1)	This macro deactivates menu structure items.
sc_menu_force_mobile (boolean)	This macro is used to force the creation of menus to mobile devices.
sc_menu_item	This macro identifies the menu item selected.
sc_script_name	This macro identifies the application name that was selected in the menu.

sc_actionbar_clicked_state()

The **sc_actionbar_clicked_state()** macro allows the developer to retrieve the current state of an action bar button at the moment it is clicked.
This enables validation of the button's current state and allows changes based on the system's business rules.

> This macro is exclusive to AJAX events for action bar buttons.

Examples

Example 1: Identifying the current state of the button in an AJAX event to toggle its state when clicked.

```
if (sc_actionbar_clicked_state() == 'pending') {
    sc_actionbar_state('ajax_btn', 'sent');
} else {
    sc_actionbar_state('ajax_btn', 'pending');
}
```

Example 2: Displaying the current state of the button.

```
echo sc_actionbar_clicked_state();
```

Macro Scope

Grid application
onClick

sc_actionbar_disable("button_name")

The **sc_actionbar_disable()** macro disables an action bar button, preventing it from being clicked.

This macro can be used for any type of button (link or AJAX) on the action bar.

Parameter Definition

Parameter	Syntax	Description
button_name	**Variables are not allowed.** The button name must be specified using **double** or **single quotes**. **Example:** sc_actionbar_disable("ajax_btn");	**This parameter is mandatory.** Specify the name of the action bar button to disable it.

Examples

Example 1: Basic macro usage

```
sc_actionbar_disable("link_detalhe");
```

Example 2: Disabling the link button to the order detail if there are no items in the order.

```
// Example created in the onRecord event of the Grid using the "order" and "orderdetail" tables
// If no items are in the order, the button will be disabled.

$query = "SELECT count(*) FROM OrderDetails WHERE OrderID = " . {OrderID};
sc_lookup(ds, $query);

if ({ds[0][0]} == 0) {
   // Example of using the sc_actionbar_disable macro
   sc_actionbar_disable("button_name");
} else {
   sc_actionbar_enable("button_name");
}
```

Macro Scope

Grid application
onClick onRecord

sc_actionbar_enable("boton_nombre")

Enables the action bar button, making it available for clicking.

This macro can be used for any type of button (link or ajax) on the action bar.

Parameter definition

Parameter	Syntax	Description
button_name	**It is not possible to use variables**, the name of the button must be informed using **double quotes** or **single quotes** **Example** sc_actionbar_enable("ajax_btn");	**This parameter is mandatory.** We must inform the name of the action bar button that will be enabled.

Examples

Example 1: Basic macro usage

```
sc_actionbar_enable("button_name");
```

Example 2: Enabling the link button to the order detail if there is any record to be displayed.

```
// Example created in the onRecord event of the grid using the order table and orderdetail table
// If there are no items ordered in the measurement, the button will be disabled

$query = "SELECT count(*) FROM OrderDetails WHERE OrderID =" . {OrderID};
sc_lookup(ds, $query);

if ({ds[0][0]} <> 0) {

   //Example of using the sc_actionbar_enable macro
   sc_actionbar_enable("button_name");

} else {

   sc_actionbar_disable("button_name");

}
```

Macro Scope

Grid application
onClick onRecord

sc_actionbar_hide("button_name")

This macro hides the button created on the action bar, which can be displayed again with the sc_actionbar_show macro.

This macro can be used with any type of button (link or ajax) created in the action bar.

Parameter definition

Parameter	Syntax	Description
button_name	**It is not possible to use variables.** The button name must be informed using **double quotes** or **single quotes**. **Example** sc_actionbar_hide("ajax_btn");	**This parameter is mandatory.** We must inform the name of the action bar button that must be hidden.

Exemplos

Example 1: Basic macro usage

```
sc_actionbar_show("button_name");
```

Example 2: Showing the order detail link button if there are any records to show.

```
// Example created in the onRecord event of the query using the order and orderdetail table
// If there are no items in the order, the detail link button will be hidden.

$query = "SELECT count(*) FROM OrderDetails WHERE OrderID =" . {OrderID};
sc_lookup(ds, $query);

if ({ds[0][0]} == 0) {

    //Example of using the macro sc_actionbar_hide
    sc_actionbar_hide("button_name");

} else {

    sc_actionbar_show("button_name");

}
```

Macro Scope

Grid application
onClick onRecord

sc_actionbar_show("button_name")

This macro displays the action bar button, hidden using the sc_actionbar_hide macro

This macro can be used for any type of button (link or ajax) on the action bar.

Parameter definition

Parameter	Syntax	Description
button_name	**It is not possible to use variables.** The button name must be informed using **double quotes** or **single quotes**. **Example** sc_actionbar_show("ajax_btn");	**This parameter is mandatory.** We must inform the name of the action bar button that will be displayed.

Exemplos

Example 1: Basic macro usage

sc_actionbar_show("button_name"**);**

Example 2: Showing the order detail link button if there are any records to show.

```
// Example created in the onRecord event of the query using the order and orderdetail table
// If there are no items in the order, the button will not be displayed

$query = "SELECT count(*) FROM OrderDetails WHERE OrderID =" . {OrderID};
sc_lookup(ds, $query);

if ({ds[0][0]} <> 0) {

   //Exemplo de utilização da macro sc_actionbar_enable
   sc_actionbar_show("button_name");

} else {

   sc_actionbar_hide("button_name");

}
```

Macro Scope

Grid application
onClick onRecord

sc_actionbar_state("button_name", "state_name")

This macro allows you to change the current state of the action bar button through.

To use the button's, ajax event, we recommend using this macro together with sc_actionbar_clicked_state(); that makes it possible to recover the current state of the button.

Definition of Parameters

Parameters	Type	Values	Description
button_name	string	The button name must be informed using **double quotes** or single quotes. Example In bold, you can see the example of how the parameter button_name must be informed: sc_actionbar_state(**"ajax_btn"**, "sent");	**This is a mandatory parameter.** You need inform the name of the button created on the action bar.
state_name	string	The state name must be informed using **double quotes** or single quotes. Example In bold, you can see the example of how the parameter state_name must be informed: sc_actionbar_state("ajax_btn", **"sent"**);	**This is a mandatory parameter.** You need inform the name of the next state that the button will assume.

Examples

Example - Changing the *ajax_btn* button state between **pending** and **sent** in the button's onclick event.

```
if (sc_actionbar_clicked_state() == 'pending') {

      sc_actionbar_state('ajax_btn', 'sent');

} else {

      sc_actionbar_state('ajax_btn', 'pending');

}
```

Example - Changing the state of the *ajax_btn* button between **pending** and **sent**, and storing the change in the base in the *email_sent* field.

```
if (sc_actionbar_clicked_state() == 'pending') {

 $update = "update orders set email_sent = 'S' WHERE orderid =". {orderid};
 sc_exec_sql($update);
      sc_actionbar_state('ajax_btn', 'sent');

} else {

 $update = "update orders set email_sent = 'N' WHERE orderid =". {orderid};
 sc_exec_sql($update);
      sc_actionbar_state('ajax_btn', 'pending');

}
```

Example - Display of the state of the *ajax_btn* button according to the value stored in the *email_sent* ield in the *onRecord* event of the Grid application.

```
if ({email_sent} == 'S') {

   sc_actionbar_state('ajax_btn', 'sent');

} else {

   sc_actionbar_state('ajax_btn', 'pending');

}
```

Macro Scope

Grid application
onClick onRecord

sc_ajax_javascript('JavascriptMethodName', array("parameter"))

Watch a tutorial

This macro allows the execution of JavaScript methods called from some events from **form, control, and calendar applications.**

To use this macro you need to create a javascript method in the Programming settings.

This macro receives two parameters:

The firs parameter (required) receive the javascript method name.
The second (optional) receive an array with the method parameters on it.

Examples:

Ex. 1: Method javascript without parameters:
sc_ajax_javascript('method_name');
sc_ajax_javascript('sum');

Ex. 2: Method javascript with a parameter:
sc_ajax_javascript('method_name', array("parameter"));

Ex. 3: Method javascript with two parameters:
sc_ajax_javascript('method_name', array("param1","param2"));

Example using global variables:
sc_ajax_javascript('sum', array([num1],[num2]));

Macro Scope

calendar application	Form application	Control Form
ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onValidate onValidateFailure onValidateSuccess	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onValidateFailure

sc_ajax_message("Message", "Title", "Parameters", "Parameters_Redir", "String_toast");

Watch a tutorial

The sc_ajax_message() macro allows the display of customized messages during the execution of applications using Ajax events.

Through this macro, it is possible to present alerts, confirmations, or notifications using modal windows, toast-style notifications, and automatic redirection options.

Its use is exclusive to Ajax events.

Parameters

Parameter	Required	Accepts Empty	Description and example
message	Yes	No	The message text to be displayed. May contain HTML. **Example:** • "Operation completed successfully!" • "Error saving data"
title	No	Yes	The message window title. If not provided, it will be displayed without a title. **Example:** • "Attention" • "Information"
configString	No	Yes	Defines the behavior and style of the message. It must be passed as a string with multiple options separated by &. **See details in Table 2: Configuration String** **Example:** • "toast=Y&timeout=3&type=success&toast_pos=top-end" • "modal=Y&button=Y&button_label=OK"

Parameter	Required	Accepts Empty	Description and example
redirectParams	No	Yes	Defines the values that will be sent to the application specified in redir, in the format field1=value1;field2=value2. **Example:** "client_id=123;client_name=JOHN"

Even though not mandatory, the parameters must be passed in the correct order. If an intermediate parameter is omitted, it must be kept as an empty value.

Table 2: Configuration String

Property	Description	Values	Example	Restrictions / Notes
modal	Defines if the message will be displayed in a modal.	**Y** - Message will open in a Modal **N** - Message will display without a modal. This property can be omitted.	• "**modal=Y**" • "**modal=Y**&width=50"	Only in applications with **SweetAlert disabled**.
timeout	Defines how long the message is displayed in seconds. Only works with messages without a confirmation button.	Accepts only numbers	• "**timeout=5**" • "toast=Y&**timeout=5**" • "button=N&**timeout=5**"	Not compatible with confirmation button • In applications with **SweetAlert disabled**, requires **button=N** or omitted. • In applications with **SweetAlert enabled**, requires **toast=Y** or omitted.
button	Controls the display of the confirmation button inside the message. The behavior is to confirm the action and close the message.	**Y** - Confirmation button will be shown. **N** - Message will show without a confirmation button. It will respect the *timeout*. Omitting the property has the same effect.	• "button=Y" • "button=Y&button_label=Confirm"	Only in applications with **SweetAlert disabled**.
button_label	Defines the label text shown on the button	Button text	• "button=Y&button_label=Close" • "button=Y&button_label=OK"	Requires confirmation button • Only in applications with **SweetAlert disabled** • Requires **button=Y**.
top	Sets the vertical position of the message using the *top* CSS property. If omitted, position is calculated automatically.	Accepts only numbers	• "top=650" • "top=650&timeout=5"	Only in applications with **SweetAlert disabled** and requires **modal=N** or omitted.
left	Sets the horizontal position of the message using the *left* CSS property. If omitted, position is calculated automatically.	Accepts only numbers	• "left=650" • "top=50&left=650&timeout=5"	Only in applications with **SweetAlert disabled** and requires **modal=N** or omitted.
width	Sets the message box width in pixels	Accepts only numbers	• "width=650" • "width=650&timeout=5"	Only in applications with **SweetAlert disabled**
height	Sets the message box height in pixels	Accepts only numbers	• "height=350" • "modal=Y&height=350&width=650"	Only in applications with **SweetAlert disabled**

Property	Description	Values	Example	Restrictions / Notes
redir	URL or application name for redirection. This attribute works only if the button property is enabled.	String with the application name or URL.	• "redir=form_comparison" • "button=Y&redir=form_comparison" • "redir=https://scriptcase.com.br/&target=_blank"	Requires confirmation button: • Application with **SweetAlert disabled** requires *button=Y* or omitted. • Application with **SweetAlert enabled** requires *toast=N* or omitted.
redir_target	Defines how the link specified in the redir parameter will be opened.	**_blank** - Opens in a new window or tab. **_self** - Opens in the same frame it was clicked (default behavior). **_parent** - Opens in the parent frame. **_top** - Opens occupying the full window body (breaks all frames).	• "redir=https://scriptcase.com.br/&target=_blank" • "redir=form_comparison&target=_parent"	Only if *redir* is defined.
show_close	Displays a close button on the title bar. This property, when enabled, forces the title bar to be displayed even if the title parameter is empty.	**Y** - Close button is shown **N** - Close button is not shown	"show_close=Y"	Only in applications with **SweetAlert disabled**
body_icon	Displays or hides the message icon. Even when this property is enabled, icon display depends on the application's Theme configuration.	**Y** - Message icon will be displayed. **N** - Message icon will not be displayed.	"body_icon=N"	Only in applications with **SweetAlert disabled**
toast	Displays the message as a toast	**Y** - Uses toast notification **N** - Uses SweetAlert message	"toast=Y"	Only in applications with **SweetAlert enabled**
toast_pos	Sets the toast position on the screen	• top • top-start • top-end • center • center-start • center-end • bottom • bottom-start • bottom-end	"toast_pos=center&toast=Y&timeout=5" "toast_pos=top&toast=N&type=warning"	Only in applications with **SweetAlert enabled** and requires **toast=Y** or omitted.
type	Defines the icon to be displayed in the message	• warning • error • success • info • question	"toast=Y&timeout=5&type=success" "toast=N&type=warning"	Only in applications with **SweetAlert enabled**

Example

1. Message without SweetAlert using Modal

```
sc_ajax_message("Data saved successfully!", "Confirmation", "modal=Y&button=Y&button_label=OK&show_close=Y");
```

2. Message without SweetAlert

sc_ajax_message("Attention! Please fill in all fields.", "Warning", "button=Y");

3. Toast Message

sc_ajax_message("Product added to cart!", "", "toast=Y&type=success&toast_pos=top-end&timeout=3");

4. Message using SweetAlert

sc_ajax_message("Error processing request!", "Error", "type=error&button=Y");

To display the message using Toast, the "Use SweetAlert" option must be enabled in the application interface.

Macro Scope

calendar application	Grid application	Form application	Control Form
ajaxFieldonBlur Onchange OnClick OnFocus	onClick	ajaxFieldonBlur Onchange OnClick OnFocus	onBlur onChange onClick onFocus

sc_ajax_refresh

The sc_ajax_refresh() macro is used to reload the data in a grid application without the need to refresh the entire page.

Its use is exclusive to AJAX events in the grid, whether in the onClick event of a field or in the action bar.

Example of use

```
$upd = "UPDATE order SET status = 'completed' WHERE id =". {orderid};
sc_exec_sql($upd);
// Now the macro will be used to refresh the application's data.
sc_ajax_refresh();
```

Macro Scope

Grid application
onClick

sc_alert("Message", $array)

Watch a tutorial

This macro sends a message in the alert style of Javascript (Sweetalert) or Sweetalert2.

Ps

To display the message using a sweetalert2 library, the Use SweetAlert option must be enabled in the application where the macro will be used.

This option can be enabled in the application's configuration menu. To access the click on: Application> Configuration the option is in the Notification Settings block

Check below the parameters supported by the macro:

- **Message:** Mandatory parameter that contains the message that will be displayed.
- **$array:** This parameter is optional when using the macro, more mandatory when we want to use Sweetalert2.
  When used, it expects to receive an array with Sweetalert2's settings.

Check the list of configurations accepted by Sweetalert2 in this macro:

Parameter	Value	Description
title	Example of how it should be informed within the array: **'title' => 'Título'**	Text that will be displayed in the card title, if the toast option is marked as false. 'toast' => false
type	Example of how it should be informed within the array: **'type' => 'warning'** List of accepted options: • **success** • **error** • **warning** • **info**	Defines the type of message that will be displayed.
timer	Example of how it should be informed within the array: **'timer' => '2000'**	Defines the time in milliseconds that the message will be displayed.
showConfirmButton	Example of how it should be informed within the array: **'showConfirmButton' => false** List of accepted options: • **True:** Exibe o botão de confirmação. • **false:** Não exibe o botão de confirmação.	Defines whether a confirmation button is displayed.
position	Example of how it should be informed within the array: **'position' => 'bottom-end',** List of accepted options: • **top** • **top-start** • **top-end** • **center** • **center-start** • **center-end** • **bottom** • **bottom-start** • **bottom-end**	Defines the position where the pop-up will be displayed.
toast	Example of how it should be informed within the array: **'toast' => true**	Defines whether or not we will use toast to display messages. When enabled, this option does not display the confirmation button.

Check below the examples of use of the macro:

***E.g.*: Message using Sweetalert**

```
sc_alert("This is an alert message!!");
```

***E.g.*: Message using Sweetalert2**

```
$params = array(
'title' => 'Title',
'type' => 'success',
'timer' => '2000',
'showConfirmButton' => false,
'position' => 'bottom-end',
'toast' => true
);

sc_alert("Inserted successfully!", $params);
```

Macro Scope

Blank application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu

onExecute	onApplicationInit onFooter onHeader onScriptInit	onFooter onHeader onScriptInit onNavigate onRecord	onApplicationInit onFilterInit	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onLoad	onApplicationInit onExecute onLoad

sc_api_download(profile, settings, file, destination)

This macro is used to download the files stored in the Storage services available in Scriptcase, they are: Google Drive, DropBox and Amazon S3.

First of all, it is necessary to create the keys to use these APIs, either for use with the macro or through the tool interface.

See below for how to obtain the credentials for each of them.

- Click here to learn how to get Google Drive credentials
- Click here to learn how to get DropBox credentials
- Click here to learn how to get Amazon S3 credentials

Parameters

Parameters	Value	Description
profile	Example of how the parameter should be informed. *'profile' => 'profile_name',* This parameter accepts only the name of the profile created. Click here, to see how to create a profile.	We must inform the name of the profile creating in the APIs option in the Tools menu. When used, this parameter loads the settings informed when the profile is created. **When used, the settings parameter does not need to be informed.** For more details see **Using the macro with profile**
settings	Example of how the parameter should be informed. *'settings' => [* *'app_name' => 'scriptcase',* *'gateway' => 'dropbox',* *'api_key' => '',* *'api_secret' => '',* *'access_token' => '',* *],* **Note:** For more information about the parameter, see the exclusive settings table below, for each of the APIs	It receives an array with the access credentials for the API that will be used. **When used, the profile parameter does not need to be entered.**
file	Example of how the parameter should be informed. *'file' => {image_file_name},* Allowed options: • {image_file_name} • [global_variable] • 'File path'	This parameter receives the full path until the file is located. When using a field, it must be the upload nome do file type.
destination	Example of how the parameter should be informed. **Using a field** *'parents' => {field_name},* **Informando uma String** *'parents' => 'folder_name',* Allowed options: • {field_name} • [global_variable] • 'Folder name or local cloud storage path'	Directory where the file will be stored When not informed, the file will be stored at the root of the storage server.

Using the macro with profile

The use of the macro with a profile requires the prior creation of the same.

Click here to see how to create a profile on each of the available storage APIs.

The profile contains the data to use the desired API, thus, when using a profile in the macro, it is not necessary to use the settings parameter.

> The profile parameter accepts only the name of the profile created, for the dynamic use of the macro, it will be necessary to use the settings parameters.

Example of using the macro with the profile.

```
sc_api_download([
  'profile' => 'myApi',
  'file' => {file},
  'destination' => {destination},
]);
```

Using the macro with settings parameter

The array with the access credentials that must be passed in the settings parameter varies according to the API used.

See below how to use this parameter according to each of the available APIs.

> When using the settings parameter in the configuration in the macro, it is not necessary to inform a profile.

Using The Macro With Settings - Google Drive API

First, we must have the credentials for using the Google Drive API in hand, click here to learn how to obtain them.

After configuring the credentials, see below all the values of the array that must be passed in the settings parameter using the Google Drive API.

All values below are mandatory when using the settings parameter

Index	Value	Description
app_name	Example of how the parameter should be informed. *'app_name' => 'scriptcase',* This index accepts only string	App name entered when creating access credentials.
gateway	Example of how the parameter should be informed. *'gateway' => 'google_driver',* This index accepts only string	Gateway name to identify the API used.
json_oauth	Example of how the parameter should be informed. *'json_oauth' => 'JSON_code',* Allowed options: • 'string' • {field_name} • $variable	Receives the contents of the user generated authentication (JSON) file. This file must be downloaded after creating the credentials on the google website.
auth_code	Example of how the parameter should be informed. *'auth_code' => '5/4wHt7TBTY4MtvQ',* Allowed options: • {field_name} • $variable	Receives the auth_code code, generated after configuring the API following the steps described in the tutorial How to generate the auth_code for using the Google Driver API
token_code	Example of how the parameter should be informed. *'token_code' => '4/4wHt7TBTY4MtvQ',* Allowed options: • 'string' • {field_name} • $variable	It is possible to generate the token_code using our sc_api_gc_get_obj macro

Examples when using the macro in Google Drive implementing the parameter settings

```
sc_api_download([
  'settings' => [
    'app_name' => {appname},
    'gateway' => 'google_drive',
    'json_oauth' => {jsonoauth},
    'auth_code' => {authcode},
    'token_code' => {tokencode},
  ],
'file' => {files},
'destination' => {destination},
]);
```

Using A Macro With Settings - Dropbox

First, we must have the credentials for using the Google Drive API in hand, click here to learn how to obtain them.

After configuring the credentials, see below all the values of the array that must be passed in the settings parameter using the Dropbox API.

All values below are mandatory when using the settings parameter

Index	Value	Description
app_name	Example of how the parameter should be informed. *'app_name' => 'scriptcase',* This index accepts only string	App name entered when creating access credentials.

gateway	Exemplo de como o parâmetro deve ser informado. *'gateway' => 'google_driver',* This index accepts only string	Gateway name to identify the API used.
api_key	Example of how the parameter should be informed. *'api_key' => {field_name},* Allowed options: • 'string' • {field_name} • $variable	Receives the app key, generated after configuring the API.
api_secret	Example of how the parameter should be informed. *'api_secret' => {field_name},* Allowed options: • 'string' • {field_name} • $variable	Receives the app secret, generated after configuring the API.

Examples when using the macro in Dropbox implementing the parameter settings

```
sc_api_download([
  'settings' => [
    'gateway' => 'dropbox',
    'api_key' => {apikey},
    'api_secret' => {apisecret},
    'access_token' => {accesstoken},
  ],
'file' => {file},
'destination' => {destination},
]);
```

Using The Macro With Settings - Amazon S3

First, we must have the credentials for using the Amazon S3 STORAGE API in hand, click here to learn how to obtain them.

After configuring the credentials, see below all the values of the array that must be passed in the settings parameter using the Amazon S3 API.

All values below are mandatory when using the settings parameter]

Index	Value	Description
app_name	Example of how the parameter should be informed. *'app_name' => 'scriptcase',* This index accepts only string	App name entered when creating access credentials.
gateway	Example of how the parameter should be informed. *'gateway' => 'google_driver',* This index accepts only string	Gateway name to identify the API used.
api_key	Example of how the parameter should be informed. *'api_key' => {nome_campo},* Allowed options: • 'string' • {field_name} • $variable	Receives the api key, generated after configuring the API.
api_secret	Example of how the parameter should be informed. *'api_secret' => {nome_campo},* Allowed options: • 'string' • {field_name} • $variable	Receives the api secret, generated after configuring the API.
region	Example of how the parameter should be informed. *'region' => 'us-east-1',* The region parameter of the macro accepts only the region code. To make it easier, follow a link to check the code for each region. https://docs.aws.amazon.com/general/latest/gr/rande.html	Receives the region code, the same used in the creation of credentials.
bucket	Example of how the parameter should be informed. *'bucket' => 'bucket_name',* Allowed options: • 'string' • {field_name} • $variable	Receives the Bucket Name, generated after configuring the API.

Examples when using the macro in Amazon S3 implementing the parameter settings

```
sc_api_download([
'settings' => [
'app_name' => 'scriptcase',
    'gateway' => 'S3',
    'api_key' => {apikey},
    'api_secret' => {apisecret},
```

```
    'region' => {api_region},
    'bucket' => {api_bucket},
  ],
'file' => {file},
'destination' => {destination},
]);
```

Macro Scope

Grid application	Form application
onRecord	onLoad onLoadRecord

sc_api_gc_get_obj($app_name , $json_oauth, $auth_code)

The **sc_api_gc_get_obj** macro generates the token_code.
It has three parameters:

1. **app_name** - Receive the app_name, available in your api configuration. This parameter accepts strings or variable.
2. **json_oauth** - Receives the content of the json file, this file is available for download in your API settings. This parameter accepts strings or variable.
3. **auth_code** - Receives the auth_code generated in the previous step or another valid auth_code that you have already generated.

For more usage and example information visit our knowledge base article: How to generate the auth code and token code

Macro Scope

calendar application	Grid application	Search application	Form application	Control Form	Menu application
onApplicationInit onScriptInit	onScriptInit	onFilterInit	onScriptInit	onApplicationInit onScriptInit	onApplicationInit

sc_api_gc_get_url($app_name, $json_oauth)

The **sc_api_gc_get_url** macro generates a URL for google account user authentication used for API configuration.
It has two parameters:

1. app_name - Receive the app_name, available in your api configuration. This parameter accepts strings or variable.
2. json_oauth - Receive the content of the json file, this file is available for download in your api settings. This parameter accepts strings or variable.

For more usage and example information visit our knowledge base article: How to generate the auth code and token code

Macro Scope

calendar application	Form application	Control Form	Menu application	ReportPDF application
onApplicationInit onScriptInit	onScriptInit	onScriptInit	onApplicationInit	onApplicationInit

sc_api_storage_delete(profile, file, parents)

The macro sc_api_storage_delete is used to delete files stored on the storage services available in Scriptcase: Google Drive, Amazon, and Dropbox.

To use the macro, you need to configure the APIs; see the documentation below.

- Google Drive API Configuration
- Dropbox API Configuration
- Amazon S3 API Configuration

Parameter List

Parameter	Example	Description
profile	In this parameter, you should provide the profile name configured in the API menu. *'profile' => 'my_api',*	This receives the profile name created in the **Tools > API** menu, retrieving the data entered for API usage. Click here to see how to create a profile.
file	**Using a field** *'file' => {upload_file_name},* The file name can be provided using: • {upload_file_name} • [global_variable] • 'File path string'	Receives the name of the file to be deleted from the storage server.
parents	**Using a field** *'parents' => {text_field_name},* The directory can be provided using: • {text_field_name} • [global_variable] • 'Folder name or local path in cloud storage'	Directory where the file is stored in the cloud. If no value is provided, the API will use the root directory of the storage server to find the file to be deleted.

Example

```
sc_api_storage_delete([
    'profile' => 'my_profile',
    'file' => {file_name},
    'parents' => 'storage_directory',
]);
```

Macro Scope

sc_api_upload(profile, settings, file, parents)

This macro is used to upload files to the cloud using the Google, Amazon and DropBox Storage APIs.

First of all, it is necessary to create the keys for using the API, either with the macro or in the fields interface.

- Click here to learn how to get Google Drive credentials
- Click here to learn how to get DropBox credentials
- Click here to learn how to get Amazon S3 credentials

After obtaining the credentials, proceed with the macro configuration.

See the list of parameters below.

Parameters	Value	Description
profile	Example of how the parameter should be informed. *'profile' => 'profile_name',* This parameter accepts only the name of the profile created. Click here, to see how to create a profile.	We must inform the name of the profile creating in the APIs option in the Tools menu. When used, this parameter loads the settings informed when the profile is created. **When used, the settings parameter does not need to be informed.** For more details see **Using the macro with profile**

settings	Example of how the parameter should be informed. *'settings' => [* *'app_name' => 'scriptcase',* *'gateway' => 'dropbox',* *'api_key' => '',* *'api_secret' => '',* *'access_token' => '',* *],* **Note:** For more information about the parameter, see the exclusive settings table below, for each of the APIs	It receives an array with the access credentials for the API that will be used. **When used, the profile parameter does not need to be entered.**
file	Example of how the parameter should be informed. *'file' => {image_file_name},* Allowed options: • {image_file_name} • [global_variable] • 'File path'	This parameter receives the full path until the file is located. When using a field, it must be the upload nome do file type.
parents	Example of how the parameter should be informed. **Using a field** *'parents' => {field_name},* **Informando uma String** *'parents' => 'folder_name',* Allowed options: • {field_name} • [global_variable] • 'Folder name or local cloud storage path'	Directory where the file will be stored When not informed, the file will be stored at the root of the storage server.

As seen above, the **settings** parameter receives an array with the access credentials for the API used.

> This parameter should be used only when we have not defined a profile in the Tools> API menu

Each API has its own settings for the settings parameter, see below how each should be configured.

Using the macro with profile

The use of the macro with a profile requires the prior creation of the same.

Click here to see how to create a profile on each of the available storage APIs.

The profile contains the data to use the desired API, thus, when using a profile in the macro, it is not necessary to use the settings parameter.

> The profile parameter accepts only the name of the profile created, for the dynamic use of the macro, it will be necessary to use the settings parameters.

Example of using the macro with the profile.

```
sc_api_upload([
  'profile' => 'profile_name',
  'file' => {file},
  'parents' => 'folder_name',
]);
```

Using the macro with settings parameter

The array with the access credentials that must be passed in the settings parameter varies according to the API used.

See below how to use this parameter according to each of the available APIs.

> When using the settings parameter in the configuration in the macro, it is not necessary to inform a profile.

Using the macro with settings - Google Drive API

First, we must have the credentials for using the Google Drive API in hand, click here to learn how to obtain them.

After configuring the credentials, see below all the values of the array that must be passed in the settings parameter using the Google Drive API.

All values below are mandatory when using the settings parameter

Index	Value	Description
app_name	Example of how the parameter should be informed. *'app_name' => 'scriptcase',* This index accepts only string	App name entered when creating access credentials.
gateway	Example of how the parameter should be informed. *'gateway' => 'google_driver',* This index accepts only string	Gateway name to identify the API used.

json_oauth	Example of how the parameter should be informed. *'json_oauth' => 'JSON_code',* Allowed options: • 'string' • {field_name} • $variable	Receives the contents of the user generated authentication (JSON) file. This file must be downloaded after creating the credentials on the google website.
auth_code	Example of how the parameter should be informed. *'auth_code' => '5/4wHt7TBTY4MtvQ',* Allowed options: • {field_name} • $variable	Receives the auth_code code, generated after configuring the API following the steps described in the tutorial How to generate the auth_code for using the Google Driver API
token_code	Example of how the parameter should be informed. *'token_code' => '4/4wHt7TBTY4MtvQ',* Allowed options: • 'string' • {field_name} • $variable	It is possible to generate the token_code using our sc_api_gc_get_obj macro

Examples when using the macro in Google Drive implementing the parameter settings

```
sc_api_upload([
  'settings' => [
     'app_name' => 'scriptcase',
     'gateway' => 'google_drive',
     'json_oauth' => '',
     'auth_code' => '',
     'token_code' => '',
  ],
  'file' => {file},
  'parents' => '',
]);
```

Using a macro with settings - Dropbox

First, we must have the credentials for using the Google Drive API in hand, click here to learn how to obtain them.

After configuring the credentials, see below all the values of the array that must be passed in the settings parameter using the Dropbox API.

All values below are mandatory when using the settings parameter

Index	Value	Description
app_name	Example of how the parameter should be informed. *'app_name' => 'scriptcase',* This index accepts only string	App name entered when creating access credentials.
gateway	Exemplo de como o parâmetro deve ser informado. *'gateway' => 'google_driver',* This index accepts only string	Gateway name to identify the API used.
api_key	Example of how the parameter should be informed. *'api_key' => {field_name},* Allowed options: • 'string' • {field_name} • $variable	Receives the app key, generated after configuring the API.
api_secret	Example of how the parameter should be informed. *'api_secret' => {field_name},* Allowed options: • 'string' • {field_name} • $variable	Receives the app secret, generated after configuring the API.

Examples when using the macro in Dropbox implementing the parameter settings

```
sc_api_upload([
  'settings' => [
     'app_name' => 'scriptcase',
     'gateway' => 'dropbox',
     'api_key' => '',
     'api_secret' => '',
     'access_token' => '',
  ],
  'file' => {file},
  'parents' => '',
]);
```

Using the macro with settings - Amazon S3

First, we must have the credentials for using the Amazon S3 STORAGE API in hand, click here to learn how to obtain them.

After configuring the credentials, see below all the values of the array that must be passed in the settings parameter using the Amazon S3 API.

All values below are mandatory when using the settings parameter

Index	Value	Description
app_name	Example of how the parameter should be informed. *'app_name' => 'scriptcase',* This index accepts only string	App name entered when creating access credentials.
gateway	Example of how the parameter should be informed. *'gateway' => 'google_driver',* This index accepts only string	Gateway name to identify the API used.
api_key	Example of how the parameter should be informed. *'api_key' => {nome_campo},* Allowed options: • 'string' • {field_name} • $variable	Receives the api key, generated after configuring the API.
api_secret	Example of how the parameter should be informed. *'api_secret' => {nome_campo},* Allowed options: • 'string' • {field_name} • $variable	Receives the api secret, generated after configuring the API.
region	Example of how the parameter should be informed. *'region' => 'us-east-1',* The region parameter of the macro accepts only the region code. To make it easier, follow a link to check the code for each region. https://docs.aws.amazon.com/general/latest/gr/rande.html	Receives the region code, the same used in the creation of credentials
bucket	Example of how the parameter should be informed. *'bucket' => 'bucket_name',* Allowed options: • 'string' • {field_name} • $variable	Receives the Bucket Name, generated after configuring the API.

Examples when using the macro in Amazon S3 implementing the parameter settings

```
sc_api_upload([
   'settings' => [
      'app_name' => 'scriptcase',
      'gateway' => 'S3',
      'api_key' => '',
      'api_secret' => '',
      'region' => '',
      'bucket' => '',
   ],
   'file' => {file},
   'parents' => '',
]);
```

Macro Scope

Form application	Control Form
onAfterDelete onAfterUpdateAll onBeforeInsertAll onValidate	onValidate

sc_apl_conf("Application", "Property", "Value")

Watch a tutorial

The sc_apl_conf() macro allows one application to dynamically modify execution properties of another application. For example, it can force a Form to start in insert mode, make a Grid start with the Search screen, define the number of rows or columns, or enable/disable toolbar buttons.

> **Note:** This macro must be used from one application to modify the behavior of another.

Form Application Properties

Property	Value	Description
start	new	Starts the Form in insert mode.
insert	on/off	Enables or disables the "Add New" button.
update	on/off	Enables or disables the "Save" button.
delete	on/off	Enables or disables the "Delete" button.
field_display_off	field	Hides a specific field dynamically.
field_display_on	field	Shows a specific field dynamically.
field_readonly	field	Sets a field to read-only dynamically.
rows	number	Defines the number of rows per page (for multiple records Form).
rows_ins	number	Defines the number of insert rows (for multiple records Form).

Grid Application Properties

Property	Value	Description
start	filter	Starts the Grid using the Search screen.
cols	number	Defines the number of columns (vertical or slide layout).
rows	number	Defines the number of rows per page.
lig_edit	on/off	Enables or disables the edit icon (pencil) for record editing.

Common Property (All Applications)

Property	Value	Description
exit	app/URL	Redirects the application to a specified location on exit.

Examples

```
// Example 1: Start "my_form" in insert mode
sc_apl_conf("my_form", "start", "new");

// Example 2: Disable record insertion in "my_form"
sc_apl_conf("my_form", "insert", "off");

// Example 3: Make "my_field" in "my_form" read-only
sc_apl_conf("my_form", "field_readonly", "my_field");

// Example 4: Show "my_field" dynamically in "my_form"
sc_apl_conf("my_form", "field_display_on", "my_field");

// Example 5: Start "my_grid" using the Search screen
sc_apl_conf("my_grid", "start", "filter");

// Example 6: Set "my_grid" to show 20 rows
sc_apl_conf("my_grid", "rows", "20");

// Example 7: Hide "my_field" dynamically in "my_form"
sc_apl_conf("my_form", "field_display_off", "my_field");

// Example 8: Trigger configuration via toolbar menu button
if ({sc_menu_item} == "btn_1") {
  sc_apl_conf("form_customer", "start", "new");
}
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tr
onExecute	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	on on on

sc_apl_default("application", "type");

Watch a tutorial

The sc_apl_default() macro allows you to define, in the initial application, what should happen when the user session is lost.

Parameters

- **apl**: Specifies the application to which the user will be redirected after the session is lost.
- **type**: Defines the redirect type. Accepted values:
  - **R**: Redirects immediately to the specified application.
  - **M**: Displays a message before redirecting.

Example

In a login application set as the initial application, we can define that the user should be redirected to the same login screen after session loss:

```
sc_apl_default("app_login", "R");
```

Macro Scope

calendar application	chart application	Grid application	Search application	Form application	Control Form	ReportPDF application
onCalendarScriptInit onScriptInit	onScriptInit	onScriptInit	onFilterInit	onScriptInit	onScriptInit	onScriptInit

sc_apl_status("Application", "Status")

Watch a tutorial

The sc_apl_status() macro is used to control application access. It dynamically enables or disables applications based on user permissions or custom logic.

Parameters

- **application**: The name of the application, or a variable containing the name, to be enabled or disabled.
- **status**: The value or variable that defines the application's status. Accepted values:
  - "on": Enables the application.
  - "off": Disables the application.

Examples

```
// Example 1: Disable a specific application
sc_apl_status("atu_cad", "off");

// Example 2: Dynamic control using variables
sc_apl_status({var_name_appl}, {var_status});
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsiv Menu
onExecute	onApplicationInit onCalendarApplicationInit onCalendarScriptInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit onValidate onValidateFailure onValidateSuccess	onApplicationInit onLoad	onApplicati onLoad

sc_appmenu_add_item("Menu_Name", "Id_Item", "Id_Parent", "Label", "Aplication", "Parameters", "Icon", "Hint", "Target","mega menu")

Watch a tutorial

The sc_appmenu_add_item macro dynamically adds items to the menu application initialized by the sc_appmenu_create macro.

> It is important to note that to create a dynamic menu using macros, the application must not contain any items created in the Menu Items interface.

Syntax

```
// Initializes the dynamic menu. This call is required before using sc_appmenu_add_item
sc_appmenu_create("menu_app_name");

// Adds an item to the already initialized dynamic menu
sc_appmenu_add_item(
  "menu_app_name",
  "id_item",
  "id_item_parent",
  "label",
  "application",
  "parameters",
  "icon",
  "hint",
  "target",
  "megamenu"
);
```

Parameters

Parameter	Required	Accepts Empty	Description and example
menu_app_name	Yes	No	Receives the name of the menu application where the items will be created. Accepts fixed text in double quotes, local variable, or global variable. **Example:** • "menu" • $local_variable • [global_var]
id_item	Yes	No	Defines the ID of the created item. This parameter accepts fixed text in double quotes or variable. **Example:** • "item_01" • "item_" . $array[0]
id_item_parent	Yes	Yes	Should be informed only if the item is a subitem. In this case, use the ID of the parent item. **Example:** • "item_01" • "item_" . $array[0]
label	Yes	No	Receives the name of the item that will be displayed in the application. The value can be provided as fixed text, variable, or lang. **Example:** • "Item name" • $itemName • {lang_item_name}
application_name	Conditional[1]	Yes	Defines the application to be opened when the item is clicked. Can be left blank if the item contains subitems. [1] Can be left blank if the item is a group (parent item), without a click action to open an application.
parameters	No	Yes	String of parameters that will be passed when the item is triggered. These parameters must be retrieved as global variables. **Example:** • param1="value";param2=$variable;param3={field}
hint	No	Yes	Text that will be displayed when hovering over the item. Can be fixed text, variable, or lang. **Example:** • "Item name" • $itemName • {lang_item_name}
icon	No	Yes	Defines the Font Awesome icon (or image) for the menu item. The value must be enclosed in double quotes. **Example:** • "fa fa-user"
target	No	Yes	Defines how the link will be opened. **Example:** • "_self" – Same window • "_blank" – New tab • "_parent" – Opens in the parent window context
megamenu	No	Yes	Defines whether the item will be displayed as a mega menu. To function correctly: – The item must be a top-level menu (parent), i.e., id_item_parent must be blank. – The item must contain at least two levels of subitems. **Example:** • "S" – Item will be shown as a mega menu • "" – Item will be shown as a standard menu

Examples

1. Static menu with direct values

```
sc_appmenu_create("menu");

// Main fixed item
```

```
sc_appmenu_add_item("menu", "item_0", "", "Home", "app_home", "", "fa fa-home", "Home page", "_self", "");

// Group: Records
sc_appmenu_add_item("menu", "item_1", "", "Records", "", "", "fa fa-folder", "", "", "");
sc_appmenu_add_item("menu", "item_2", "item_1", "Clients", "app_clients", "", "fa fa-users", "", "_self", "");
sc_appmenu_add_item("menu", "item_3", "item_1", "Suppliers", "app_suppliers", "", "fa fa-truck", "", "_self", "");
sc_appmenu_add_item("menu", "item_4", "item_1", "Products", "app_products", "", "fa fa-box", "", "_self", "");

// Group: Reports
sc_appmenu_add_item("menu", "item_5", "", "Reports", "", "", "fa fa-file", "", "", "");
sc_appmenu_add_item("menu", "item_6", "item_5", "Billing", "app_rel_billing", "", "fa fa-chart-line", "", "_self", "");
sc_appmenu_add_item("menu", "item_7", "item_5", "Stock", "app_rel_stock", "", "fa fa-warehouse", "", "_self", "");
sc_appmenu_add_item("menu", "item_8", "item_5", "Clients", "app_rel_clients", "", "fa fa-address-book", "", "_self", "");
```

2. Using array and foreach

```
sc_appmenu_create("menu");

// Home
sc_appmenu_add_item("menu", "item_0", "", "Home", "app_home", "", "fa fa-home", "Home page", "_self", "");

$groups = array(
  array(
    "id"     => "item_1",
    "label"  => "Records",
    "icon"   => "fa fa-folder",
    "items"  => array(
      array("label" => "Clients",     "app" => "app_clients",     "icon" => "fa fa-users"),
      array("label" => "Suppliers",   "app" => "app_suppliers",   "icon" => "fa fa-truck"),
      array("label" => "Products",    "app" => "app_products",    "icon" => "fa fa-box")
    )
  ),
  array(
    "id"     => "item_5",
    "label"  => "Reports",
    "icon"   => "fa fa-file",
    "items"  => array(
      array("label" => "Billing",     "app" => "app_rel_billing",   "icon" => "fa fa-chart-line"),
      array("label" => "Stock",       "app" => "app_rel_stock",     "icon" => "fa fa-warehouse"),
      array("label" => "Clients",     "app" => "app_rel_clients",   "icon" => "fa fa-address-book")
    )
  )
);

$id_counter = 2;

foreach ($groups as $group) {
  sc_appmenu_add_item("menu", $group["id"], "", $group["label"], "", "", $group["icon"], "", "", "");

  foreach ($group["items"] as $item) {
    $id = "item_" . $id_counter++;
    sc_appmenu_add_item("menu", $id, $group["id"], $item["label"], $item["app"], "", $item["icon"], "", "_self", "");
  }
}
```

3. Using foreach with database lookup

```
sc_appmenu_create("menu");

// Home
sc_appmenu_add_item("menu", "item_0", "", "Home", "app_home", "", "fa fa-home", "Home page", "_self", "");

// Query with JOIN
sc_lookup(menu_data, "
  SELECT
    g.id AS group_id,
    g.name AS group_name,
    i.id AS item_id,
    i.name AS item_name,
    i.app AS app,
    i.icon AS icon
  FROM groups g
  JOIN items i ON i.group_id = g.id
  WHERE g.active = 'Y' AND i.active = 'Y'
  ORDER BY g.id, i.id
");

$created_groups = array();

if (isset({menu_data[0][0]})) {
  foreach ({menu_data} as $row) {
    $group_id   = "item_" . $row[0];
    $group_name = $row[1];
    $item_id    = "item_" . $row[2];
    $item_name  = $row[3];
    $item_app   = $row[4];
    $item_icon  = $row[5];

    if (!in_array($group_id, $created_groups)) {
      sc_appmenu_add_item("menu", $group_id, "", $group_name, "", "", "fa fa-folder", "", "", "");
      $created_groups[] = $group_id;
    }

    sc_appmenu_add_item("menu", $item_id, $group_id, $item_name, $item_app, "", $item_icon, "", "_self", "");
  }
}
```

4. Mega menu example

```
sc_appmenu_create("menu");

// Mega menu root
sc_appmenu_add_item("menu", "item_0", "", "Management", "", "", "fa fa-sitemap", "", "", "S");

// Mega menu groups
$groups = array(
  array(
    "id"    => "item_1",
    "label" => "Records",
    "items" => array("Clients", "Products", "Users")
  ),
  array(
    "id"    => "item_2",
    "label" => "Finance",
    "items" => array("Accounts Payable", "Revenue", "Billing")
  ),
  array(
    "id"    => "item_3",
    "label" => "Reports",
    "items" => array("Sales", "Stock", "Clients")
  )
);
```

```
$id_counter = 4;

foreach ($groups as $group) {
  sc_appmenu_add_item("menu", $group["id"], "item_0", $group["label"], "", "", "", "", "", "");

  foreach ($group["items"] as $label) {
    $id  = "item_" . $id_counter++;
    $app = "app_" . strtolower(str_replace(" ", "_", $label));
    sc_appmenu_add_item("menu", $id, $group["id"], $label, $app, "", "", "", "_self", "");
  }
}
```

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_appmenu_create("Menu_Name")

Watch a tutorial

The sc_appmenu_create() macro initializes the array used to dynamically build a menu application. The name of the target menu must be passed as a parameter.

Note: This macro must be used in Control application events or other application types, but not in the **onLoad** event of the Menu application itself.

Example

```
// Example 1: Creates a dynamic menu structure named "menu_main"
sc_appmenu_create("menu_main");
```

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_appmenu_exist_item("Menu_Name", "Id_Item")

Watch a tutorial

The sc_appmenu_exist_item() macro checks if a menu item exists in a dynamic menu structure previously created with sc_appmenu_create(). It returns a boolean value: true if the item exists, or false otherwise.

Parameters

Parameter	Description
Menu_Name	Name of the menu application.
Id_Item	Identifier of the menu item to check.

Example

```
// Example: Check if item_2 exists in "menu_main"
// If it exists, update it; otherwise, add it

if (sc_appmenu_exist_item("menu_main", "item_2")) {
  sc_appmenu_update_item("menu_main", "item_2", "item_1", "Category Form", "form_category");
} else {
  sc_appmenu_add_item("menu_main", "item_2", "item_1", "Category Form", "form_category");
}
```

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_appmenu_remove_item("Menu_Name", "Id_Item")

Watch a tutorial

The sc_appmenu_remove_item() macro removes a menu item dynamically from a menu previously created with sc_appmenu_create().

Parameters

Parameter	Description
menu_name	Name of the menu application.
id_item	Identifier of the menu item to be removed.

Example

```
// Example 1: Remove item_2 from the "menu_main"
sc_appmenu_remove_item("menu_main", "item_2");
```

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_appmenu_reset("Menu_Name")

Watch a tutorial

The sc_appmenu_reset() macro resets the internal array used in the dynamic creation of a menu application. It clears all items previously added through macros like sc_appmenu_add_item().

This macro expects the name of the menu application as its only parameter.

Example

```
// Example 1: Resets the structure of the "menu_main" application
sc_appmenu_reset("menu_main");
```

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_appmenu_update_item("Menu_Name", "Id_Item", "Id_Parent", "Label", "Aplication", "Parameters", "Icon", "Hint", "Target")

Watch a tutorial

Updates an application menu item dynamically

It is only possible to use this macro when there is an array created before by the macro *sc_appmenu_create*.

It is only possible to use this macro on items created before by the macro *sc_appmenu_add_item*.

Parameter	Description
menu_name	Application menu item name
id_item	Application menu item id
id_parent	Application menu item parent id. For roots, the null/empty value is passed.
Label	Application menu item description.
Application	Menu item URL or application name
Parameters	String with a parameter to pass to the application menu item Ex.: param1=valor;param2=valor
Icon	Icon path used in the menu item.
Hint	Descriptive Hint for the menu item.
Target	The target for menu item link

Ex. 1:

```
sc_appmenu_update_item('menu_main','item_2','item_1','Category Form','form_category');
```

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_begin_trans("Connection")

Watch a tutorial

This macro starts a set of transactions in the database.

For the use of this macro you need to have a **transactional** database.

In form applications, the events (onAfterInsert, onAfterUpdate, onAfterDelete, onBeforeInsert, onBeforeUpdate or onBeforeDelete) are automatically protected by a transaction control, since the connection is the same of the application.

In the other cases, if the user need to make a transaction control, must begin with this macro and finish with the sc_commit_trans() macro to confirm the updates or sc_rollback_trans() to cancel the transactions.

The "connection" parameter is optional, necessary only if the command is executed in a different connection from the actual application.

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onApplicati onExecute onLoad

sc_block_display(Block_Name, on/off)

Watch a tutorial

Dynamically determines the display fields of a specific block.

By default, all the blocks are displayed ("on" condition).

Ex. 1:

```
if ({type_customeri} == "personal")
{
sc_block_display(company, off);
}
else
{
sc_block_display(personal, off);
}
```

Ex 2: Using a local variable as second parameter

```
$var = "off";

sc_block_display(company, $var);
```

Obs: *In grids, this macro only works with "slide" orientation.*

Macro Scope

calendar application	chart application	Grid application	Dashboard	Form application	Control Form
onAfterInsert onAfterUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onScriptInit	onScriptInit onRecord	onScriptInit	onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onClick onScriptInit onLoadAll onRefresh

sc_btn_copy

Watch a tutorial

Available when the "Copy" button is clicked. It can be tested and used inside the ScriptCase events, allowing specific programming in run time.

Ex. 1:

```
if (sc_btn_copy)

{

sc_message("Record copied successfully!");

}
```

Macro Scope

calendar application	Form application
onScriptInit onLoad	onScriptInit onLoad

sc_btn_delete

Watch a tutorial

Available when the "Delete" button is clicked. It can be tested and used inside the ScriptCase events, allowing specific programming in run time.

Ex. 1:
if (sc_btn_delete)
{
sc_error_message("Unable to delete this record");
}

Macro Scope

calendar application	Form application
onValidate onValidateFailure onValidateSuccess	onValidate onValidateFailure onValidateSuccess

sc_btn_disable('button_id', 'on/off')

Watch a tutorial

This macro is used to disable or enable buttons from a Menu application.

Ex:

sc_btn_disable('btn_2', 'off');

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onLoad

sc_btn_disabled("name_button", "status")

Watch a tutorial

This macro is intended to dynamically enable or disable a toolbar button.

It has two required parameters:

Parameters	Value	Description
name_button	The value must be informed using variables or in string format, with double or single quotes. In bold, the example of how the parameter should be reported follows: sc_btn_disabled(**'update'**, 'off'); **Note:** The name of the buttons should be all in tiny.	This parameter sets the button that will have its display changed. Button names are available in the table below, separated by application. • Grid • Form/Calendar • Control
status	Like the field name, the parameter accepts only string, with single or double quotation marks. In bold, the example of how the parameter should be reported follows: sc_btn_disabled('update', **'off'**);	This parameter sets whether the button is enabled, when set to **on**, or disabled if set to **off** **Values accepted in the parameter:** • **on -** The button will be disabled; • **off -** The button will be enabled;

Name of Grid buttons

Buttons Name	Description
first	Goes to the first page of the grid.
back	Goes to the previous page of the grid.
forward	Goes to the next page of the grid.
last	Goes to the last page of the grid.
filter	Displays the "Filter" button.
pdf	Generates colored PDF.
det_pdf	Generates a detailed PDF.
pdfbw	Generates black and white PDF.
xls	Generates the XLS.
xml	Generates the XML.
csv	Generates the CVS.
rtf	Generates the RTF.
word	Generates the Word.
json	Generates the JSON.

print	Displays the "Print" button in the grid.
det_print	Displays the "Print" button in detail.
summary	Displays the "Summary" button in the grid.
new	Displays the "New" button. (if the application has a link for it, or if the form is running in the grid iframe)
insert	Displays the "Include" button of the form. (only for forms running in grid´s iframe)
update	Displays the "Update" button of the form. (only for forms running in grid´s iframe)
delete	Displays the "Delete" button of the form (only for forms running in grid´s iframe)
qsearch	Displays the "Quick Search" button in the grid.
dynsearch	Displays the "Dynamic Search" button in the grid.
gridsave	Displays the "Save Grid" button in the grid.
sel_col	Select columns button
sort_col	Sort Button

Name of Form buttons

Buttons Name	Description
first	Goes to the first page of the form.
back	Goes to the previous page of the form.
forward	Goes to the next page of the form.
last	Goes to the last page of the form.
new	Displays the "New" button on the form, if it's off the form will starts in insertion mode.
insert	Displays the "Include" button of the form.
update	Displays the "Update" button of the form.
delete	Displays the "Delete" button of the form.
copy	Displays the "Copy" button of the form.
pdf	Generates PDF file from a form.
print	Open the form in print mode.
dynsearch	Displays the "Dynamic Search" button in the grid.

Name of Control Buttons

Buttons Name	Description
ok	Displays the "Ok" button on the control.
exit	Displays the "Exit" button on the control.
facebook	Displays the "Facebook" button on the control.
twitter	Displays the "Twitter" button on the control.
google	Displays the "Google+" button on the control.
paypal	Displays the "PayPal" button on the control.

Example: Enabling delete button only for admin user

```
if ( [user_login] == 'admin' ) {
   sc_btn_disabled('delete', 'on');
} else {
   sc_btn_disabled('delete', 'off');
}
```

Macro Scope

calendar application	chart application	Grid application	Form application	Control Form
onAfterDelete onAfterInsert onAfterUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onScriptInit	onScriptInit	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onScriptInit onLoad onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess

sc_btn_display("Button_Name","on/off")

Watch a tutorial

Activate toolbars buttons on the application in execution time.

Button_Name: Is the variable name that contains the name of the button to be activated/deactivated.

Buttons Name (**Grid**):

group displays the "Group By" button in the grid.

Buttons Name	**Description**
first	Goes to the first page of the grid.
back	Goes to the previous page of the grid.
forward	Goes to the next page of the grid.
last	Goes to the last page of the grid.
filter	Displays the "Filter" button.
pdf	Generates colored PDF.
det_pdf	Generates a detailed PDF.
pdfbw	Generates black and white PDF.
xls	Generates the XLS.
xml	Generates the XML.
csv	Generates the CVS.
rtf	Generates the RTF.
word	Generates the Word.
json	Generates the JSON.
print	Displays the "Print" button in the grid.
det_print	Displays the "Print" button in detail.
summary	Displays the "Summary" button in the grid.
new	Displays the "New" button. (if the application has a link for it, or if the form is running in the grid iframe)
insert	Displays the "Include" button of the form. (only for forms running in grid´s iframe)
update	Displays the "Update" button of the form. (only for forms running in grid´s iframe)
delete	Displays the "Delete" button of the form (only for forms running in grid´s iframe)
qsearch	Displays the "Quick Search" button in the grid.
dynsearch	Displays the "Dynamic Search" button in the grid.
gridsave	Displays the "Save Grid" button in the grid.
sel_col	Select columns button
sort_col	Sort Button

Buttons Name (**Form**):

Buttons Name	**Description**
first	Goes to the first page of the form.
back	Goes to the previous page of the form.
forward	Goes to the next page of the form.
last	Goes to the last page of the form.
new	Displays the "New" button on the form, if it's off the form will starts in insertion mode.
insert	Displays the "Include" button of the form.
update	Displays the "Update" button of the form.
delete	Displays the "Delete" button of the form.
copy	Displays the "Copy" button of the form.
qsearch	Displays the "Quick Search" button of the form.
pdf	Generates PDF file from a form.
print	Open the form in print mode.
dynsearch	Displays the "Dynamic Search" button in the grid.

Buttons Name (**Control**):

Buttons Name	**Description**
ok	Displays the "Ok" button on the control.
exit	Displays the "Exit" button on the control.
facebook	Displays the "Facebook" button on the control.
twitter	Displays the "Twitter" button on the control.
google	Displays the "Google+" button on the control.
paypal	Displays the "PayPal" button on the control.

OBS. Remember that the button's name must be lowercase.

Ex. 1:
sc_btn_display ('new', 'off');

Ex. 2: Using a local variable as second paramenter

$var = "off";
sc_btn_display ('new', $var);

Ex. 3: If we use variables or fields ({Fields} or [Global_Variables]) as parameters its not allowed the use of the quotes or single quotes.

sc_btn_display ({variavle_button}, 'off');

Macro Scope

calendar application	chart application	Grid application	Form application	Control Form
onAfterDelete onAfterInsert onAfterUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onScriptInit	onScriptInit	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onScriptInit onLoadAll onRefresh onValidate onValidateFailure

sc_btn_insert

Watch a tutorial

Available when the "Add" button is clicked. It can be tested and used inside the ScriptCase events, allowing specific Programmation in run time.

Ex. 1:

```
if (sc_btn_insert)
{
sc_message("Record inserted successfully");

}
```

Macro Scope

calendar application	Form application
onValidate onValidateFailure onValidateSuccess	onValidate onValidateFailure onValidateSuccess

sc_btn_label("btn_name", "new_label")

Watch a tutorial

This macro is used to dynamically change the label of the buttons, whether they are created by the developer or Scriptcase's own.

For this, it is necessary to inform two mandatory parameters, 'btn_name' and 'new_label'.

Parâmetro	Valor	Descrição
btn_name	The parameter can be entered using variables (Global or Local) or string (Using Double or Single Quotes) In bold, here is the example of how the parameter should be entered: sc_btn_label(**'update'**, 'Update Record'); Values accepted in the parameter: • **String** - Protected by single or double quotes. Ex.: 'update' • **Variables - Local or global. E.g.:** $btn_update	Receives the name of the buttons that will have the label changed. Button names, which are automatically created by Scriptcase, are available below, separated by application. • Grid/Chart • Form/Calendar • Control
new_label	The parameter can be entered using variables (Global or Local) or string (Using Double or Single Quotes) sc_btn_label('update', **'Update Record'**); **Values accepted in the parameter::** • **String** - Protected by single or double quotes. **E.g.: 'Update Record'** • **Variables -** Local or global. **E.g.:** $new_label • **Lang -**variable lang. **E.g.:** {lang_btn_label}	This parameter receives the new label of the button informed in the previous parameter. In addition to string and variable, it is also possible to use a lang to define the label, enabling the translation of your application's internationalization.

Click here and see examples of using the macro.

Nombre de los botones de cuadrícula

Buttons Name	Description
first	Goes to the first page of the grid.
back	Goes to the previous page of the grid.
forward	Goes to the next page of the grid.
last	Goes to the last page of the grid.
filter	Displays the "Filter" button.
pdf	Generates colored PDF.
det_pdf	Generates a detailed PDF.
pdfbw	Generates black and white PDF.
xls	Generates the XLS.
xml	Generates the XML.
csv	Generates the CVS.
rtf	Generates the RTF.
word	Generates the Word.
json	Generates the JSON.
print	Displays the "Print" button in the grid.
det_print	Displays the "Print" button in detail.
summary	Displays the "Summary" button in the grid.
new	Displays the "New" button. (if the application has a link for it, or if the form is running in the grid iframe)
insert	Displays the "Include" button of the form. (only for forms running in grid´s iframe)
update	Displays the "Update" button of the form. (only for forms running in grid´s iframe)
delete	Displays the "Delete" button of the form (only for forms running in grid´s iframe)
qsearch	Displays the "Quick Search" button in the grid.
dynsearch	Displays the "Dynamic Search" button in the grid.
gridsave	Displays the "Save Grid" button in the grid.
sel_col	Select columns button
sort_col	Sort Button

Name of Form buttons

Buttons Name	Description
first	Goes to the first page of the form.
back	Goes to the previous page of the form.
forward	Goes to the next page of the form.
last	Goes to the last page of the form.
new	Displays the "New" button on the form, if it's off the form will starts in insertion mode.
insert	Displays the "Include" button of the form.
update	Displays the "Update" button of the form.
delete	Displays the "Delete" button of the form.
copy	Displays the "Copy" button of the form.
pdf	Generates PDF file from a form.
print	Open the form in print mode.
dynsearch	Displays the "Dynamic Search" button in the grid.

Name of Control Buttons

Buttons Name	Description
ok	Displays the "Ok" button on the control.
exit	Displays the "Exit" button on the control.
facebook	Displays the "Facebook" button on the control.
twitter	Displays the "Twitter" button on the control.
google	Displays the "Google+" button on the control.
paypal	Displays the "PayPal" button on the control.

Exemplo: Parameter with string

```
sc_btn_label('update', 'Atualizar Registro');
```

Example: Parâmetro com lang

```
sc_btn_label("update", {lang_btn_label});
```

Example:Parameter with Global variable

```
[btn_name] = "update";
[label] = "update";
sc_btn_label([btn_name], [label]);
```

Example: Parameter with local variable

```
$btn_name = "update";
$label= "update";
```

sc_btn_label($btn_name, $label);

Macro Scope

calendar application	chart application	Grid application	Form application	Control Form
onAfterDelete onAfterInsert onAfterUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onScriptInit	onScriptInit	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onScriptInit onLoad onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onScriptInit onLoadAll onRefresh onValidate onValidateFailure

sc_btn_new

Watch a tutorial

Available when the "Add New" button is clicked. It can be tested and used inside the ScriptCase events, allowing specific programming in run time.

Ex. 1:

```
if (sc_btn_new)
{
{My_Date} = date('Y/m/d');
}
```

Macro Scope

calendar application	Form application
onLoad	onLoad

sc_btn_update

Watch a tutorial

Available when the "Save" button is clicked. It can be tested and used inside the ScriptCase events, allowing specific programming in run time.

Ex. 1:

```
if (sc_btn_update)
{
sc_error_message("Record updated successfully");
}
```

Macro Scope

calendar application	Form application
onValidate onValidateFailure onValidateSuccess	onValidate onValidateFailure onValidateSuccess

sc_calc_dv(Digit, Rest, Value, Module, Weights, Type)

Watch a tutorial

Calculate verifier (checksum) digits.

Parameter	Description
Digit	Variable that will receive the calculated digit.
Rest	Variable that will receive the "rest" of the division.
Value	Value or Variable to calculate the digit.
Module	Value or Variable containing the module to be used. The default value is the module "11".
Weights	Value or Variable containing the weights to be used. The default value is the "98765432".
Type	Value or Variable containing the parcels calculation to be used: 1 to the normal sum of the values of the parcels and 2 to the sum of each algorithm of the parcels. The default value is "1".

Ex. 1:
sc_calc_dv({my_dv}, {my_rest}, 1234567, 11, 98765432, 1);
Parcels calculation: 7x2= 14 ; 6x3=18 ; 5x4=20 ; 4x5=20 ; 3x6=18 ; 2x7=14 ; 1x8=8
Parcels sum : 14 + 18 + 20 + 20 + 18 + 14 + 8 = 112
Rest of the division of 112 / 11 = 2
Digit: 11 - 2 = 9

Ex. 2:
sc_calc_dv({my_dv}, {my_rest}, 1234567, 10, 12, 2);
Parcels calculation: 7x2= 14 ; 6x1=6 ; 5x2=10 ; 4x1=4 ; 3x2=6 ; 2x1=2 ; 1x2=2
Parcels sum: 1 + 4 + 6 + 1 + 0 + 4 + 6 + 2 + 2 = 26
Rest of the division of 26 / 10 = 6
Digit: 10 - 6 = 4

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_call_api($profile, $arr_settings)

It allows us to use Scriptcase built-in APIs. We currently offer integration with the Mandrill and Amazon SES APIs.

This macro has two parameters:

Parameters	Description
$profile	Required parameter. Can be passed empty '' or contain the gateway name configured in "tools> API".
$arr_settings	This parameter is optional as long as the "profile" is not empty. When informed, this parameter must contain an Array with the Gateway used and the API key.

Example 1 - Parameter $arr_settings, array with the API information

```
$settings = array( 'settings' => ['gateway' => 'mandrill', 'api_key'=> {api_key}] );

$mandrill = sc_call_api('', $settings);
```

Example 2 - Sending email using mandrill.

```
$txt_no_tags = strip_tags({msg});

$arr_merge = array(
              'name'  => '',
              'type'  => 'to',
              'email' => 'exemplo@exemplo.com'
            );

$var_config = array( 'settings' => ['gateway' => 'mandrill', 'api_key'=> {api_key}] );

$mandrill = sc_call_api('', $var_config);
$var_msg = array(
            'from_email'                => {from_email},
            'from_name'                  => {from_name},
             'html'                     => {msg},
             'text'                     => $txt_no_tags,
             'to'                       => array($arr_merge),
             'subject'                  => {subject},
             'important'                => true,
             'auto_text'                => null,
             'auto_html'                => null,
             'inline_css'               => null,
             'metadata'                 => array('website' => 'www.scriptcase.net')
   );

$async = false;
$retorno = $mandrill->messages->send($var_msg, $async);
```

Example 3 - Sending SMS using clickatell.

```
$var_config     = array(
  'message'     => [
    'to'      => {sms_to},
    'message'     => {sms_txt},
    ],
    'settings'    => [
    'gateway'   => {gateway},
    'auth_token'  => {auth_token},
    ]
);

sc_send_sms($var_config);
```

Example 4 - Using the macro with PagSeguro

```
  $arr_settings = [
          'gateway' => 'pagseguro',
          'environment' => 'sandbox',
          'auth_email' => 'exemplo@gmail.com',
          'auth_token' => '72AE21503DDA4840BE1DC7945F6D1CE1'
              ];
sc_call_api('',$arr_settings);
```

Example 5 - Using the macro with PayPal

```
  $arr_settings = [
          'gateway' => 'paypal',
          'environment' => 'sandbox',
          'auth_email' => 'exemplo@gmail.com',
          'auth_token' => '72AE21503DDA4840BE1DC7945F6D1CE1'
              ];
sc_call_api('',$arr_settings);
```

Macro Scope

Blank application	**calendar application**	**chart application**	**Grid application**	**Search application**	**Form application**	**Control Form**	**Menu application**	**Tree menu**	**ReportPDF application**
onExecute	onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onHeader onScriptInit onRecord	onFilterInit onFilterSave onFilterValidate	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onFooter onHeader onScriptInit onRecord

sc_captcha_display("on/off")

Dynamically controls whether the **captcha/recaptcha** configured in the application is displayed.

To use the macro, the application must be previously configured with the resource.

Example

```
if ( $_SERVER['REMOTE_ADDR'] == '127.0.0.1' ) {
   sc_captcha_display("on");
} else {
   sc_captcha_display("off");
```

Macro Scope

Form application	Control Form
onScriptInit onLoad	onScriptInit onLoadAll

sc_changed({Field_Name})

Watch a tutorial

This macro is used to check if the field name has been changed.

Ex. 1:

```
if (sc_changed({my_field}))
{
Echo "Field have been changed";
}
```

Macro Scope

calendar application	Form application	Control Form
onBeforeUpdate onValidate onValidateFailure onValidateSuccess	onBeforeUpdate onValidate onValidateFailure onValidateSuccess	onRefresh

sc_change_connection("Old_Connection", "New_Connection")

Watch a tutorial

This macro dynamically change the application connections

Ex. 1: To change the connections:

```
sc_change_connection("Old_Concection01", "New_Connection01" ; "Old_Concection02", "New_Connection02");
```

Allows to change more then one application/connection in one command. Multiple connections can be passed as parameters that must be separated by a **";"**.

Ex. 2: Using variables:

```
sc_change_connection([Global_Connetion], [Test_Connection]);
```

The changes take effect in the following applications.

Ex. 3: Changing the application main connection

```
sc_change_connection({Password}, [Test] ; "My_Connection", [Global_Connection]);
```

The changes take effect in the following applications.

Ex. 4: To delete the exchange:

To recover your previous settings you can use the sc_reset_change_connection(); macro

Macro Scope

Blank application	calendar application	chart application	Grid application	Form application	Control Form	ReportPDF application
onExecute	onScriptInit onLoad	onHeader onScriptInit	onApplicationInit onScriptInit	onScriptInit onLoad onLoadRecord	onScriptInit onLoadAll	onHeader onScriptInit onRecord

sc_change_css("attribute", "value", "field_name")

This macro allows you to manipulate the CSS properties of fields and rows in the grid application.

This way, you can change the appearance of the application dynamically and with many more customization options.

Parameter List

parameters	parameter values	Description
attribute_name	The button name must be informed using **double quotes**, **single quotes** or **variables**. In this parameter is possible to use sc_badge that apply a predefined css in the text. **Example** In this example we are using the color attribute sc_change_css('**color**', '#dcfce7');	**Required Attribute.** Defines the CSS attribute that will be changed.
value	The button name must be informed using **double quotes**, **single quotes** or **variables**. When is used sc_badge at this parameter the attribute value should be one of the colors above: blue, brown, cyan, gray, green, orange, pink, purple, red, yellow and empty string If the attribute value be empty string, the badge wll be removed **Example** In this parameter we define the value for the CSS attribute it should receive, in the example we are defining the color: #dcfce7 sc_change_css('color', '**#dcfce7**');	**Required Attribute.** Defines the value that the informed attribute will receive

field_name	The button name must be informed using **double quotes**, **single quotes** or **variables**.
	
	Example
	In this example we are defining the change in the font color of the field fld_status
	
	sc_change_css('color', '#dcfce7', '**status**');
	
	Example with sc_badge
	sc_change_css('sc_badge', '**green**', 'field_name');
		Optional Attribute.
		
		Defines the field where the CSS will be applied, if no value is entered, the change will be made in all fields of the line.
		
		To use sc_badge is necessary to inform the field name, in other hand it will not apply css to any field.

Check the list of accepted colors when using sc_badge in the attribute name below.

- blue
- brow
- cyan
- gray
- green
- orange
- pink
- purple
- red
- yellow

Example

Example of changing the font color and using the attribute sc_badge

```
switch({taskstatus}) {
 case 'COMPLETED':
  sc_change_css('color', '#dcfce7', 'taskstatus');
  sc_change_css('sc_badge', 'green', 'taskstatus');
  break;

 case 'IN-PROGRESS':
  sc_change_css('color', '#1e40af', 'taskstatus');
  sc_change_css('sc_badge', 'blue', 'taskstatus');
  break;

 case 'NOT-STARTED':
  sc_change_css('color', '#991b1b', 'taskstatus');
  sc_change_css('sc_badge', 'red', 'taskstatus');
  break;
}
```

Example of the macro with css attribute:

```
switch({taskstatus}) {
 case 'COMPLETED':
  sc_change_css('color', '#166534', 'taskstatus');
  sc_change_css('background-color', '#dcfce7', 'taskstatus');
  sc_change_css('padding-right', '0.6em', 'taskstatus');
  sc_change_css('padding-left', '0.6em', 'taskstatus');
  sc_change_css('border-radius', '14px', 'taskstatus');
  sc_change_css('font-weight', 'bold', 'taskstatus');
  sc_change_css('width', 'fit-content', 'taskstatus');
  sc_change_css('display', 'block', 'taskstatus');
  sc_change_css('padding-top', '0.2em', 'taskstatus');
  sc_change_css('padding-bottom', '0.2em', 'taskstatus');
  sc_change_css('font-size', '60%', 'taskstatus');
```

```
 sc_change_css('margin', '10%', 'taskstatus');
 break;

 case 'IN-PROGRESS':
  sc_change_css('color', '#1e40af', 'taskstatus');
  sc_change_css('background-color', '#dbeafe', 'taskstatus');
  sc_change_css('padding-right', '0.6em', 'taskstatus');
  sc_change_css('padding-left', '0.6em', 'taskstatus');
  sc_change_css('border-radius', '14px', 'taskstatus');
  sc_change_css('font-weight', 'bold', 'taskstatus');
  sc_change_css('width', 'fit-content', 'taskstatus');
  sc_change_css('display', 'block', 'taskstatus');
  sc_change_css('padding-top', '0.2em', 'taskstatus');
  sc_change_css('padding-bottom', '0.2em', 'taskstatus');
  sc_change_css('font-size', '60%', 'taskstatus');
  sc_change_css('margin', '10%', 'taskstatus');
  break;

 case 'NOT-STARTED':
  sc_change_css('color', '#991b1b', 'taskstatus');
  sc_change_css('background-color', '#fee2e2', 'taskstatus');
  sc_change_css('padding-right', '0.6em', 'taskstatus');
  sc_change_css('padding-left', '0.6em', 'taskstatus');
  sc_change_css('border-radius', '14px', 'taskstatus');
  sc_change_css('font-weight', 'bold', 'taskstatus');
  sc_change_css('width', 'fit-content', 'taskstatus');
  sc_change_css('display', 'block', 'taskstatus');
  sc_change_css('padding-top', '0.2em', 'taskstatus');
  sc_change_css('padding-bottom', '0.2em', 'taskstatus');
  sc_change_css('font-size', '60%', 'taskstatus');
  sc_change_css('margin', '10%', 'taskstatus');
  break;
}
```

Macro Scope

Grid application
onClick onRecord

sc_commit_trans("Connection")

Watch a tutorial

This macro is used to confirm a transaction set in the database.

In form applications, there are events that perform transactions with the database (Ex: onAfterInsert, onAfterUpdate, onAfterDelete, onBeforeInsert, onBeforeUpdate, onBeforeDelete, etc...).
The transaction control of these events is done by Scriptcase itself if the connection provided is the same as the application.
However, if the user has to perform any type of redirection in these events, it is necessary to use the macro sc_commit_trans before the redirect to secure the transactions previously performed.

Ex: onAfterUpdate - Using redirection after update a record:

```
sc_commit_trans();


sc_redir('grid_main');
```

The "connection" parameter is optional, being necessary only if the command is executed in a different database specified to the application.

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onApplicati onExecute onLoad

sc_concat()

Watch a tutorial

This macro returns a string resulting from a definition of two or more values Can be used in the lookup of fields or events together with the macros sc_lookup() and sc_select()

Makes concat formatting according to your database.

Ex. 1: Use(Editing Lookup):

```
SELECT field1, sc_concat(field2,' - ',field3) FROM table
```

Ex. 2: Using the macro with sc_lookup:

```
sc_lookup(Dataset,"SELECT field1,sc_concat(field2, ' - ' ,field3) FROM table");
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu
onExecute	OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateSuccess	onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateSuccess	onBlur onChange onClick onFocus onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onExecute onLoad

sc_confirm("Message")

Watch a tutorial

This macro is used to confirm that the action must be really executed, preventing mistakes.

Valid only in the PHP programming code that runs when called through a toolbar button.

Parameter

This macro has only one parameter, which is the confirmation message that will be displayed.

Example:
sc_confirm("Do you really wish to execute this action??");

Related Links
See a practical example of how to use the sc_confirm macro

Macro Scope

Form application	Control Form
onClick	onClick

sc_connection_edit("Connetion_Name", $arr_conn)

Watch a tutorial

The edited connection will only be available in the next application.

This macro edits an existing connection at runtime. ("Connection_Name", $arr_conn)

1st Parameter: Name of the connection to be edited.

2nd Parameter: Array containing the connection information to be edited. See each index below:

Index	Description	Example
['server']	Database server	$arr_conn['server'] = "127.0.0.1"
['user']	Database user	$arr_conn['user'] = "root"
['password']	Database password	$arr_conn['password'] = "secretpass123"
['database']	Database used by the connection	$arr_conn['database'] = "sc_samples"
['persistent']	Defines whether the connection is persistent	$arr_conn['persistent'] = "Y" / "N"
['encoding']	Sets the connection encoding	$arr_conn['encoding'] = "utf8"

Note: It is not mandatory to edit all the array indices; you can pass only the index you want to change.

Example of macro usage:

```
$arr_conn = array();

$arr_conn['user'] = "admin2";
$arr_conn['password'] = "admin2pass";
$arr_conn['database'] = "sc_samples2";

sc_connection_edit("sc_conexao", $arr_conn);
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu	Re ap
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarApplicationInit onCalendarScriptInit onScriptInit onLoad onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onScriptInit	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onExecute onLoad	on on on on on

sc_connection_new("Connection_Name", $arr_conn)

Watch a tutorial

When used, the macro creates a new connection with the information inserted in its parameters. This connection is available throughout the current session Scriptcase and ceases to exist when the session is closed.

The new connection will only be available in the next application.

1º Parameter: Connection name.

Note: If there is a connection created within the Scriptcase with the same name, this macro has no effect. Connections created within the Scriptcase prevail. If you want to edit an existing

connection, see the documentation for the macro **sc_connection_edit**.

2º Parameter: Array of items containing the connection information. Check out the indices of the array:

Indice	Description	Example
['drive']	Driver of the database used for the connection (see table below)	$arr_conn['drive'] = "oci8"
['server']	Database server (host)	$arr_conn['server'] = "127.0.0.1"
['user']	Database username	$arr_conn['user'] = "root"
['password']	Database password	$arr_conn['password'] = "secretpass123"
['database']	Database name used in the connection	$arr_conn['database'] = "sc_samples"
['persistent']	Defines if the connection is persistent or not	$arr_conn['persistent'] = "Y" / "N"
['encoding']	Configure the connection encoding	$arr_conn['encoding'] = "utf8"

Note: It is required that all items are filled, with the exception of items **['persistent']** and **['encoding']**.

See below the driver's list:

Driver	Description
access	MS Access
ado_access	MS Access ADO
odbc	ODBC Generic
db2	DB2
db2_odbc	DB2 ODBC Native
odbc_db2	DB2 ODBC Generic
odbc_db2v6	DB2 ODBC Genérico 6 or less
pdo_db2_odbc	DB2 PDO ODBC
pdo_ibm	DB2 PDO
firebird	Firebird
pdo_firebird	Firebird PDO
borland_ibase	Interbase 6.5 or more
ibase	Interbase
pdo_informix	Informix PDO
informix	Informix
informix72	Informix 7.2 or less
ado_mssql	MSSQL Server ADO
pdo_sqlsrv	MSSQL Server NATIVE SRV PDO
mssqlnative	MSSQL Server NATIVE SRV
odbc_mssql	MSSQL Server ODBC
mssql	MSSQL Server
pdo_dblib	DBLIB
pdo_mysql	MySQL PDO
mysqlt	Mysql (Transactional)
mysql	MySQL (Non-Transactional)
pdo_oracle	Oracle PDO
oci805	Oracle 8.0.5 or more
odbc_oracle	Oracle ODBC
oci8	Oracle 8
oci8po	Oracle 8 Portable
oracle	Oracle 7 or less
postgres7	PostgreSQL 7 ou Acima
pdo_pgsql	PostgreSQL PDO
postgres64	PostgreSQL 6.4 or more
postgres	PostgreSQL 6.3 or more
pdosqlite	SQLite PDO
sqlite	SQLite
sybase	Sybase
pdo_sybase_dblib	Sybase PDO DBLIB
pdo_sybase_odbc	Sybase PDO ODBC

Example:

```
$arr_conn = array();

$arr_conn['drive'] = "mysqlt";
$arr_conn['server'] = "127.0.0.1";
$arr_conn['user'] = "root";
$arr_conn['password'] = "pass123";
$arr_conn['database'] = "sc_samples";
$arr_conn['persistent'] = "Y";
$arr_conn['encoding'] = "utf8";

sc_connection_new("new_conn_mysql", $arr_conn);
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu	Re ap
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarApplicationInit onCalendarScriptInit onScriptInit onLoad onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onScriptInit	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onExecute onLoad	on on on on on

sc_date(Date, "Format", "Operator", D, M, Y)

Watch a tutorial

This macro has the aim to calculate the increment or decrement of dates. For its use are necessary some parameters as follows:

Parameter	Description
Date	Date field that contains the date to be modified.
Format	Formatting thats the date field is.
Operator	"+" for increment and "-" for decrement.
D	Number of days to increment or decrement.
M	Number of months to increment or decrement.
Y	Number of Years to increment or decrement.

Ex. 1:
{birthdate} = sc_date({birthdate}, "dd/mm/yyyy", "+", 30, 0, 0);

Ex. 2:
$new_date = sc_date({birthdate}, "yyyy-mm-dd", "-", 15, 3, 2);

Ex. 3:
{inclusion_dt} = sc_date(date('Ymd'), "yyyymmdd", "-", 0, 1, 1);

OBS: *To be certified of the date format, before use the function, try it in any event: echo "form_date =". {name of the field date};*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_date_conv({Field_Date}, "Input_Format", "Output_Format")

Watch a tutorial

This macro converts the date field passed as a parameter with an input format to another field with an output format.

Check below the list of parameters for this macro and the accepted values.

Parameter	Values	Description
field_name	Example of how the parameter should be informed: **{field_name}**	This parameter must receive the name of the data field that will have the manipulated value
input_format	The string must have the same format used in the database. Example of how to enter the date string in the macro: **"yyyy-mm-dd"** Accepted values: ▪ **Date string -** For example "**yyyy-dd-mm**" ▪ **db_Format** - Retrieves the date format from the database.	Receives the string with the current date format, saved in the database. If the format informed is different from the one used in the database, the macro will not work as expected.
output_format	The String must have the format in which the date will be displayed. Example of how to enter the date string in the macro: "**yyyy/dd/mm/**" Accepted options ▪ **Date string** - For example "**yyyy-dd-mm**" ▪ **sc_format_region** - Value formatted according to the regional configuration ▪ **db_Format -** Convert the date to the database's native value	Receive the string with the desired date display format. If you want to use the format according to the regional configuration, the value **sc_format_region** in this parameter must be informed

For more information on the data strings consult the PHP Manual

Ex. 1: Converts the date from**"dd/mm/yyyy"** format to the **"yyyymmdd"** format**.**

```
{field_date} = sc_date_conv({field_date}, "dd/mm/aaaa", "aaaammdd");
```

Ex. 2: Converts the date from the **database native format** to **"dd/mm/yyyy"** format.

```
{field_date} = sc_date_conv({field_date}, "db_format", "dd/mm/aaaa");
```

Ex. 3: Converts the date from the **"dd/mm/yyyy"** format to the Database native format(**db_format**)**.**

```
{field_date} = sc_date_conv({field_date}, "dd/mm/aaaa", "db_format");
```

Ex. 4: Converting the date from the database's native format (**db_format**) to the regional setting format (**sc_format_region**)

```
{field_date} = sc_date_conv({field_date},"db_format","sc_format_region");
```

Ex. 5: Converting the date from the **"yyyy-mm-dd"** to the regional Setting format (**sc_format_region**)

```
{field_date} = sc_date_conv({field_date},"aaaa-mm-dd","sc_format_region");
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	ReportPDF application
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit onRecord

sc_date_dif({Date1}, "Format Date1", {Date2}, "Format Date2")

Watch a tutorial

Calculates the difference between two dates (passed as parameters) in days.

Parameter	Description
Date1	Date1 value or variable.
Format Date1	Value or Variable containing the date format stored on Date1.
Date2	Date2 value or variable.
Format Date2	Value or Variable containing the date format stored on Date2.

Ex. 1:
{amount_days} = sc_date_dif({date1}, "aaaa-mm-dd", {date2}, "mm/dd/aaaa");

Ex. 2:
{amount_days} = sc_date_dif("2000-05-01", "aaaa-mm-dd", "04/21/2004", "mm/dd/aaaa");
{amount_days} would be equal to -1451 (days)

Note: *Formats can be different but must be in days, months and years.*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_date_dif_2({Date1}, "Format Date1", {Date2}, "Format Date2", Option)

Watch a tutorial

Calculates the difference between two dates (passed as parameters) returning the number of days, months, and years.

The result is returned in an array structure, where index 0 have the number of days, index 1 has the number of months and the index 2 have the number of years.

Parameter	Description
Date1	Date1 value or variable.
Format Date1	Value or Variable containing the date format stored on Date1.
Date2	Date2 value or variable.
Format Date2	Value or Variable containing the date format stored on Date2.
Option	1 = Doesn't consider the initial day; 2 = Considers the initial day.

Ex. 1:
$diferences = sc_date_dif_2 ({date1}, "yyyy-mm-dd", {date2}, "mm/dd/yyyy", 1);
{dif_days} = $diferences[0];
{dif_months} = $diferences[1];
{dif_years} = $diferences[2];

Ex. 2:
$diferences = sc_date_dif_2 ("2000-05-01", "yyyy-mm-dd", "04/21/2004", "mm/dd/yyyy", 1);
$diferences[0] = 20 (days)
$diferences[1] = 11 (months)
$diferences[2] = 3 (years)

Ex. 3: Using the option 2
$diferences = sc_date_dif_2 ("2000-05-01", "yyyy-mm-dd", "04/21/2004", "mm/dd/yyyy", 2);
$diferences[0] = 21 (days)
$diferences[1] = 11 (months)
$diferences[2] = 3 (years).

Note: *Formats between the two dates can be different but it must have days, months and years. You also need to know the internal format being returned for the macro to work correctly.*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_date_empty({Field_Date})

Watch a tutorial

This macro checks if a date field is empty.

Ex. 1:
```
if (sc_date_empty({my_date}) )
{
sc_error_message("Invalid Date");
}
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	ReportPDF application
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onHeader	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit	onFilterRefresh onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onLoadAll onRefresh	onHeader onRecord

sc_decode({My_Field})

Watch a tutorial

This macro returns the encrypted field or variable to its original value.

Ex. 1: Using a local variable:
```
{my_field} = sc_decode($field_crypt);
```

Ex. 2: Using an user variable:
```
$my_var = sc_decode($field_crypt);
```

Note: Scriptcase's owner, *Netmake, isn't responsible by the cryptography algorithm integrity (it is an open-source algorithm)*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_encode({My_Field})

Watch a tutorial

Returns the field or variable with the content encrypted.

Ex. 1: Encrypting a local variable:
$field_crypt = sc_encode({my_field});

Ex. 2: Encrypting an user variable:
$field_crypt = sc_encode($my_var);

Note: *Netmake isn't responsible by the encrypting algorithm integrity, (The algorithm provided is open source).*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_error_continue("Event")

Watch a tutorial

This macro deactivates the Scriptcase standard database error treatment message for the event passed as parameter allowing it to be substituted by the programmer customized error treatment.

It can be used in along with variable: {sc_erro_delete}, {sc_erro_insert} and/or {sc_erro_update}

The events to be passed as parameter are: insert, delete or update.

Ex. 1: Using on the event **onDelete** for the message returned by mssql, trigger treatment, which cancels the record exclusion, displays a error message in the format [Microsoft] [ODBC SQL Server Driver] [SQL Server] You can't erase the record.

```
sc_error_continue("delete");
if (!empty({sc_error_delete}))
{
$tmp = strrpos({sc_error_delete}, "]");
if ($tmp !== false)
{
{sc_error_delete} = substr({sc_error_delete}, $tmp 1);
}
sc_error_message({sc_error_delete});
sc_error_exit;
}
```

Macro Scope

calendar application	Form application	Control Form
onBeforeDelete onBeforeInsert onBeforeUpdate	onBeforeDelete onBeforeInsert onBeforeUpdate	onScriptInit

sc_error_delete

Watch a tutorial

This variable returns a string with a database error message, which occurs by trying to exclude a record.

It is used when there is a need to treat these messages.

Like for example, in case of database validations via triggers or procedures that they return messages.

Ex. 1: Content of the variable **{sc_error_delete}**, returning an error message generated by database trigger using

MsSQL.“[Microsoft] [ODBC SQL Server Driver] [SQL Server] You can't erase this register.”

Note: To access the database return error, sees the macro "**sc_error_continue**".

Macro Scope

calendar application	Form application
onAfterDelete	onAfterDelete

sc_error_exit(URL / My_Application, "Target");

Watch a tutorial

The sc_error_exit macro interrupts code execution.

When used without a parameter, the macro uses a return, so when using it in a PHP method only the execution of the code in the method will be interrupted, returning to the originating event and executing the rest of the code. In this case, to interrupt the execution of the application event, the macro must also be used when returning the method.

This macro must be used after the macro sc_error_message();

Parameters

It has two parameters that are not mandatory and the use or not of the parameters interferes with the functioning of the macro.

Parameter	Description	Parameter passing

app_name/URL	**This parameter is not mandatory.**
	
	When informed, whenever the macro is triggered, the user will be redirected to the informed application or page, which can be opened in the same tab or in a new one,**according to the target parameter**.
		The application name or URL must be enclosed in double or single quotation marks.
		
		Example with URL
		sc_error_exit('www.scriptcase.com.br');
		
		Example with Application
		sc_error_exit('app_Login');

target	**This parameter is not mandatory.**
	
	It defines whether the application/URL will be opened in the same tab or in a new one.
	
	Values can be:
	• **_blank** - Which displays the application/URL in a new tab.
	• **_self** - Displays the application/URL in the same tab as the application.
	
	If no value is entered, the **_self** will be used as the default.
		This parameter must be entered in double or single quotes.
		
		Example opening the same tab
		sc_error_exit('www.scriptcase.com.br');
		
		Example opening the same tab with target
		sc_error_exit('www.scriptcase.com.br', '_self');
		
		Example opening new tab
		sc_error_exit('app_Login', '_blank');

Examples

Ex. 1: Displays only the error message.

```
sc_error_message("Error message");
sc_error_exit();
```

Ex. 2: Displays error messages and the "OK" button to redirect to the specified URL.

```
sc_error_message("Error message");
sc_error_exit(http://www.meusite.com);
```

Ex. 3: Displays error messages and the "OK" button to redirect to the informed Application.

```
sc_error_message("Error message");
sc_error_exit("my_application", "_blank");
```

Macro Scope

calendar application	Grid application	Search application	Form application	Control Form
ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onValidate onValidateFailure onValidateSuccess	onClick	onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onNavigate onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess

sc_error_insert

Watch a tutorial

This variable returns a string with the database error, that occurs when adding a record. It is used when is needed to treat these messages. For example, in the case of database validations via triggers or procedures that returns a message.

Ex. 1: Variable **{sc_erro_insert}**, returning an error message generated by database trigger the using MS SQL

“[Microsoft] [ODBC SQL Server Driver] [SQL Server] You cannot add this record.”

Note: *To access the return of the database error, see the macro "**sc_error_continue**".*

Macro Scope

calendar application	Form application
onAfterInsert	onAfterInsert

sc_error_message("Text")

Watch a tutorial

This macro generates an error message or error messages.

In "**Form and Control**" applications, the messages are presented together with other error messages found in the application, except when using the "**sc_error_exit**" macro.

In "**Grid and Menu**" applications the messages only will be presented through the "**sc_error_exit**" macro.

Ex. 1:
```
if ({discount} > 0.10 && [glo_usr] == 'operator')
{
sc_error_message("Discounting of " . {discount} . " above of the allowed one");
}
```

In this example, if the value in the discount field is over 0.10 and the user try to add or save this record, the operation is finished and the error message is presented. The command must finish with ");" (close parentheses and semicolon) used as a delimiter for the macro interpreter.

Macro Scope

Blank application	calendar application	Search application	Form application	Control Form
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onValidate onValidateFailure onValidateSuccess	onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onLoad onNavigate onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess

sc_error_update

Watch a tutorial

This variable returns a string with a database error message that occurs when updating a record. It is used when there is a need to customize these messages. For example, in the case of database validations via triggers or procedures that return message.

Ex. 1: Variable **{sc_error_update}**, returning an error message generate by a database trigger using MS SQL.

“[Microsoft][ODBC SQL Server Driver][SQL Server] You can't update this record.”

Note: *To have access to the return of database error see the macro "**sc_error_continue**".*

Macro Scope

calendar application	Form application
onAfterUpdate	onAfterUpdate

sc_event_hint('field_name', 'help message', maximum_width)

The sc_event_hint macro allows the developer to define a help text (tooltip) for fields containing links created from the onClick event.

The macro is not available for events created from action bar buttons.

Parameters

The macro has three parameters, two of which are mandatory (field_name and message) and one optional (maximum_width). The parameters must be separated by a comma (,).

Parameter	Description	Example

field_name	**Mandatory parameter** that receives the field name, as a string, where the help text (tooltip) will be displayed.
	
	The field name must be enclosed in double or single quotes, containing only the field name. **This parameter does not accept the use of variables.**
		// Example of parameter definition.
		sc_event_hint(**'field_name'**, 'Click for more information', 100);

Parameter	Description	Example

message	**Mandatory parameter** that receives the help text to be displayed when hovering over the link created in the Ajax event. The text will be displayed only in the field specified in the first parameter if an onClick event is configured.
	
	The parameter can be defined using:
	
	• String (enclosed in double or single quotes)
	• PHP variable
	• Global variable
	• Lang
		// Defining the text using a **string**.
		sc_event_hint('field_name', **'Click for more information'**, 100);
		
		// Defining the text using a **PHP variable**.
		sc_event_hint('field_name', **$var_php**, 100);
		
		// Defining the text using a global variable.
		sc_event_hint('field_name', **[var_global]**, 100);
		
		// Defining the text using a **Scriptcase lang**.
		sc_event_hint('field_name', **{lang_name}**, 100);

maximum_width	Optional parameter that defines the maximum width of the hint where the message will be displayed. When reaching the defined limit, the text will wrap to the next line.
	
	The size is defined in pixels and should only contain numbers.
		// Defining the text **with a width limit**.
		sc_event_hint('field_name', 'Click for more information'**, 100**);

Macro Scope

Grid application
onScriptInit
onRecord

Example of use

Example 1: Using a lang

```
sc_event_hint("customerid", {lang_name});
```

sc_exec_sql("SQL Command", "Connection")

Example 2: Using a PHP variable.

```
$var_msg_help = "Field help message";

sc_event_hint("customerid", &var_msg_help );
```

Example 3: With validation

```
if ( {person_type} == "F" ) {

   sc_event_hint(field,"Individual");

} else {

  sc_event_hint(field,"Company");

}
```

This macro allows to condition the circumstances thats the SQL commands are executed.

The "Connection" parameter is optional. Required only, if the command is executed in a database different from the application.

Ex. 1:
if (sc_after_delete) {
sc_exec_sql("**delete** from mytable **where** key = {key_ant}");
}

Macro Scope

Blank application	**calendar application**	**chart application**	**Grid application**	**Dashboard**	**Search application**	**Form application**	**Control Form**	**Menu application**	**Responsive Menu**	**Tree me**
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarScriptInit onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onApplica onExecut onLoad

sc_exit(Option)

Watch a tutorial

The macro sc_exit defines the application's behavior after executing a **PHP** or **RUN** button. This behavior varies depending on the parameter provided.

Its use is exclusive to buttons of type **PHP** or **RUN**

Parameter

Parameters	Examples	Description
No parameter	*sc_exit();*	The confirmation screen with the "OK" button will not be displayed, and the user will be redirected to the previous application (if it exists).
ok	*sc_exit(**ok**);*	The confirmation screen with the "OK" button will be displayed, and the user will be redirected to the previous application (if it exists).
sef	*sc_exit(**sef**);*	The confirmation screen with the "OK" button will not be displayed, and the user will return to the current application, executing the SELECT command of the application again.
ref	*sc_exit(**ref**);*	The confirmation screen with the "OK" button will not be displayed, and the user will return to the current application, refreshing the data with a **refresh**.
ok, sef	*sc_exit(**ok,sef**);*	The confirmation screen with the "OK" button will be displayed, and the user will return to the current application, executing the SELECT command of the application again.
ok, ref	*sc_exit(**ok,ref**);*	The confirmation screen with the "OK" button will be displayed, and the user will return to the current application, refreshing the data with a **refresh**.

Parameters for forms and control

The options below are for transaction control and are exclusively for PHP buttons in control and form applications.

Parameters	Examples	Description
No parameter	*sc_exit();*	Does not affect database transactions.
c	*sc_exit(**c**);*	Performs pending transactions.
r	*sc_exit(**r**);*	Does not perform pending transactions.

Example

Example of using sc_exit in a RUN button.

```
// Using in the onRecord event of the RUN button.
$update = "update orders set shipaddress = 'completed' where orderid = ". {orderid};
sc_exec_sql($update);
sc_exit(ref);
```

Macro Scope

calendar application	Dashboard	Form application	Control Form
onScriptInit	onScriptInit	onClick onScriptInit	onClick onScriptInit

sc_field_color("Field", "Color")

Watch a tutorial

This macro is used to change/restore **a grid field** text color dynamically.

Ex. 1: Change the grid text field “customer_id” color.
sc_field_color ("customerid", "#33FF66");

Ex. 2: Change the "value_order” field text color, if the value is greater that 1000. Otherwise, use the default color.

```
if ({value_order} > 1000)
{
sc_field_color ("value_order", "#33FF66");
}
else
{
sc_field_color ("value_order", "");
}
```

Ex. 3: Using a local variable.
sc_field_color ("customerid", {fld_color});

Ex. 4: Using a global variable.
sc_field_color ("customerid", [glo_color]);

Macro Scope

Grid application	ReportPDF application
onScriptInit onRecord	onScriptInit onRecord

sc_field_disabled("Field_Name = True/False", "Parameter")

Watch a tutorial

This macro is used to block a field to get any data that would be typed on it.

Fields that is going to be blocked should be followed by the option "true" (default) or "false".

The parameter is optional, "I" it's used only to block de addition of new records, "U" to block only the update, in case of the parameter has not been informed the Scriptcase will use both options.

Ex. 1: Blocks only one field for addition and update.
sc_field_disabled("Field_01");

Ex. 2: Unblocking a field for addition and update.
sc_field_disabled("Field_02=false");

Ex. 3: Blocks several fields, only for an update.
sc_field_disabled("Field_01; Field_02; Field_03", "U");

Ex. 4: Combination of the block and unblock of several fields, for addition and update.
sc_field_disabled("Field_01=true; Field_02=false; Field_03=true");

Macro Scope

calendar application	Form application	Control Form	ReportPDF application
onLoad	onLoad onNavigate	onLoadAll	onHeader

sc_field_disabled_record("Field_Name = True/False", "Parametre")

Watch a tutorial

This macro has the objective to block the typing on determining fields in the lines of the Multiple Record form, Editable Grid, and Editable Grid View.

Fields that are supposed to be blocked should be applied the options, "true" (To block) or "false" (to unblock). If not informed, the default value is "true".

The optional parameter is using the “I” to block while inserting a new record and “U” to block while updating a record. If not informed, Scriptcase will block on both occasions by default.

Ex. 1: Blocking the field while including and updating.
sc_field_disabled_record(“Field_01");

Ex. 2: Unblocking the field while including and updating.
sc_field_disabled_record(“Field_02=false");

Ex. 3: Blocking the various fields only when updating.
sc_field_disabled_record(“Field_01; Field_02; Field_03", "U");

Ex. 4: Combination of blocking and unblocking various fields while including and updating.
sc_field_disabled_record("Field_01=true; Field_02=false; Field_03=true");

The macro can also work within a condition, blocking a specific field on the record line.

Example 1:
```
if ({Filed_01} == 1)
{
sc_field_disabled_record("Field_03");
}
```

Example 2:
```
switch(trim([arr_apps][ {app_name} ]))
{
case 'form':
break;
case 'calendar':
sc_field_disabled_record("priv_export; priv_print");
break;
case 'cons':
sc_field_disabled_record("priv_insert; priv_delete; priv_update");
break;
default:
sc_field_disabled_record("priv_insert; priv_delete; priv_update; priv_export; priv_print");
break;
}
```

Macro Scope

Form application
onLoadRecord

sc_field_display({My_Field}, on/off)

Watch a tutorial

This macro will display a specific field dynamically.
By default, all the fields are displayed ("on" condition").

Ex. 1:

```
if ({tp_customeri} == "personal")
{
sc_field_display({company_name}, off);
}
else
{
sc_field_display({personal_name}, off);
}
```

Ex. 2: Using a local variable as second parameter

```
$var = "off";

sc_field_display({company_name}, $var);
```

Macro Scope

calendar application	chart application	Grid application	Form application	Control Form
ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onHeader onScriptInit	onScriptInit	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeInsertAll onBeforeUpdateAll onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onScriptInit onLoadAll onRefresh onValidate onValidateFailure

sc_field_init_off(Field1, Field2,...)

Watch a tutorial

This macro is intended to inhibit the query fields on the initial load.

Fields inhibited by this macro can be enabled by using the "Columns" button.

Ex 01:

sc_field_init_off ({orderID}, {TotalValue});

Macro Scope

Grid application
onApplicationInit

sc_field_no_validate('field_name')

The macro can only be used in the onLoad event of Form, Control and Calendar applications. It allows ignoring the validations of the fields informed in the parameter, allowing the developer to dynamically "remove" the validations according to his business rule.

> It is worth mentioning that **only the validations defined in the field configuration screen will be ignored by the macro**. Validations created by the developer in events, methods or libraries will not be considered.

Check some validations that will be ignored with the use of the macro.

- Field data types: Date, credit card, email, and zip code.
- Formatting text field values: Maximum, minimum value and characters allowed.
- Formatting numeric field values: Maximum and minimum value.
- Fields marked as mandatory.

See below for details about the macro parameter.

Macro Parameter

Parameter	Syntax	Description

field_name	The parameter expects to receive the name of the fields in quotes (single or double) or in curly brackets {field_name}. It is also possible to enter two or more fields, in which case the names must be separated by a comma.
	
	Syntax with a field
	
	// field name in quotes
	sc_field_no_validate('field_name');
	
	// field name in curly brackets
	sc_field_no_validate({field_name});
	
	Syntax with two or more fields
	
	// Two and more fields
	sc_field_no_validate({field_name1}, {field_name2});
		This parameter is mandatory and must receive the name of the fields that will have the validation disregarded.
		
		It is possible to inform more than one field, in this case, the names must be separated by a comma.

Lista de Exemplos

Removing validation from a field hidden by the macro sc_field_display.

```
sc_field_display({field1}, 'off');
sc_field_no_validate({field1});
```

Example using multiple fields in curly brackets.

```
sc_field_no_validate({field1}, {field2}, {field3});
```

Example using multiple fields with quotes.

```
sc_field_no_validate('field1', 'field2', 'field3');
```

Macro Scope

calendar application	Form application	Control Form
onLoad	onLoad onLoadRecord	onLoadAll

sc_field_readonly({Field}, on/off)

Watch a tutorial

This macro dynamically sets a form field attribute to 'ReadOnly'. Use this macro only to do it at 'runtime'. Otherwise set this parameter on / off on the form interface. section "Read-Only".

The on/off parameter its optional, used by an ajax event to define if a field is going to be read-only (On) or read-only (Off).

Ex 1: To set a field as read only even when the form is in "Addition mode":

```
if (sc_btn_new)
{
sc_field_readonly({my_field});
}
```

Ex. 2: To set a field as read only dynamically use:

```
sc_field_readonly({my_field});
```

Macro Scope

calendar application	Form application	Control Form
ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeInsertAll onBeforeUpdateAll onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onScriptInit onLoadAll onRefresh onValidate onValidateFailure

sc_field_style({My_Field}, "Background-Color", "Size", "Color", "Family", "Weight")

Watch a tutorial

The sc_field_style macro allows you to dynamically change the CSS of fields in a **Grid** application.

Parameters

This macro does not allow the use of variables in the parameters. Therefore, the parameter values (arguments) must be passed directly in the macro, in the correct order.

> Only the field_name parameter is **mandatory**. However, when "skipping" a parameter, it must be passed as an **empty string**.
> For example, to change the third parameter (font-size) without modifying the background color, the macro should be used as follows: sc_field_style({data_nasc}, '', '15px');

Parameter	Description	Example
field_name	This parameter is **mandatory** and indicates the name of the field to be styled. The field name must be enclosed in curly braces: {field_name}.	sc_field_style(**{data_nasc}**, '#33FF99', '15px', '#000000', 'Arial, sans-serif', 'bold');
background-color	Changes the background color of the field cell (column) using the CSS background-color property.	sc_field_style({data_nasc}, **'#33FF99'**);
font-size	Uses the font-size property to change the font size of the field. Just provide the desired value, such as '14px'.	sc_field_style({data_nasc}, '', **'14px'**);
color	Changes the text color of the field using the CSS color property.	sc_field_style({data_nasc}, '', '', **'#33FF99'**);
font-family	Changes the font family of the field, overriding the default font used in the application.	sc_field_style({data_nasc}, '', '', '', **'Arial, sans-serif'**);
font-weight	Changes the font weight (thickness) of the field using the CSS font-weight property.	sc_field_style({data_nasc}, '', '', '', '', **'bold'**);

Usage examples

Example 1: Expired dates with red background

```
if ({data_vencimento} < date('Y-m-d')) {
    sc_field_style({data_vencimento}, '#FF4D4D', '', '#FFFFFF');
}
```

Example 2: Highlight field with larger font

```
sc_field_style({nome_cliente}, '', '18px');
```

Example 3: Monospaced font for description

```
sc_field_style({descricao}, '', '', '', 'Courier New, monospace');
```

Example 4: Bold text for "Pending" status

```
if ({status} == 'Pendente') {
    sc_field_style({status}, '', '', '', '', 'bold');
}
```

Example 5: Visual alert for "Under Review"

```
if ({status} == 'Em Análise') {
    sc_field_style({status}, '#FFF3CD', '14px', '#856404');
}
```

Macro Scope

Grid application
onRecord

sc_foot_hide()

Watch a tutorial

This macro is intended to hide the application footer.

With no mandatory parameter, the macro is available in the following applications:

- Form
- Control
- Graphic
- Grid

In the Grid, in addition to its standard use (without parameters), it is possible to individually define the module where the display of the footer will be inhibited.

Below is the list of **unique grid parameters**

- **grid -** Hide only the footer of the grid module
- **res** or **sum -** Hide only the summary module footer
- **det -** Hides the detail module footer

> Remembering that the parameters are not mandatory, in the case of the query, if no parameter is informed, the footer display will be inhibited in all modules.

Examples

Eg. no parameter - Default use of the macro, without passing a parameter, inhibits the display of the footer in the Form, Control, Graph and in all modules of the query.

```
sc_foot_hide();
```

Below are examples of using the macro with parameter, **exclusively for use in the grid**

Eg. using the 'grid' parameter - Inhibits the visualization of the footer only in the application module.

```
sc_foot_hide('grid');
```

Eg. using the 'sum' parameter - Inhibits the display of the footer only in the module in the application summary.

```
sc_foot_hide('res'); or sc_foot_hide('sum');
```

Eg. using the 'det' parameter - Inhibits the visualization of the footer only in the module in the application summary.

```
sc_foot_hide('det');
```

Eg. using more than one parameter - Inhibits the visualization of the footer in the grid and application detail modules.

In this case the parameters must be separated by a comma.

```
sc_foot_hide('grid,det');
```

Macro Scope

chart application	Grid application	Form application	Control Form
onScriptInit	onScriptInit	onScriptInit	onScriptInit

sc_format_num({My_Field}, "Group_Symb", "Dec_Symb", "Amount_Dec", "Fill_Zeros", "Side_Neg", "Currency_Symb", "Side_Currency_Symb")

Watch a tutorial

This macro is used to format numerical values.

Parameter	Description
My_Field	Variable or field to format (The return is in the next variable).
Group_Symb	Integer values grouping symbol.
Dec_Symb	Decimals separator symbol.
Amount_Dec	Number of decimals to be displayed.
Fill_Zeros	Complete decimals with zeros (fill in) (S = yes and N = no).
Side_Neg	Negative sign position. (1 = left and 2 = right)
Currency_Symb	Monetary symbol to be displayed.
Side_Currency_Symb	Currency symbol position. (1 = left and 2 = right).

Ex. 1: Formatting an integer.
sc_format_num({my_value}, '.', '', '0', 'N', '1', '');
Input value = 001234567890
Output value = 1.234.567.890

Ex. 2: Formatting a negative value and truncating decimals.
sc_format_num({my_value}, '.', '', '0', 'N', '1', '');
Input value = 0012345678.90-
Output value = -12.345.678

Ex. 3: Formatting a value with 4 decimals and filling with zeros.
sc_format_num({my_value}, '.', ',', '4', 'S', '1', '');
Input value = 0012345678.9
Output value = 12.345.678,9000

Ex. 4: Formatting a value with 2 decimals, filling with zeros and using monetary symbol.
sc_format_num({my_value}, '.', ',', '2', 'S', '1', 'US$');
Input value = 0012345678.9
Output value = US$ 12.345.678,90

Ex. 5: Formatting a value with 2 decimals, filling with zeros.
sc_format_num({my_value}, '.', ',', '2', 'S', '1', '');
Input value = .9
Output value = 0,90

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onFooter onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_format_num_region({My_Field}, "Qtde_Dec", "Insert_Zeros", "Monetary_Sym")

Watch a tutorial

This macro has the objective to format numbers, using the regional settings

Parameter	Description
My_Field	Variable or field that has a value to be formatted (if the value is 0 the result will be the same as the variable).
Qtde_Dec	Amount of decimals that'll be displayed.
Insert_Zeros	Complete with the non existing decimals (Y = yes and N = no).
Monetary_Sym	Display monetary symbol (Y = yes and N = no).

Ex. 1: Formatting a value with 2 decimal spaces, inserting the zeros and the monetary value.
sc_format_num_region({my_value}, "2", "Y", "Y");
Input value = 9
Output value = R$0,90

Macro Scope

Blank application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu
onExecute	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_form_show'on' or 'off'

Watch a tutorial

This macro dynamically display or hide a form.
By default, the form is displayed ("on" condition).

Ex. 1: Does not display the form.
sc_form_show = 'off';

Macro Scope

calendar application	Form application	Control Form
onScriptInit onLoad onValidate	onScriptInit onLoad onValidate	onScriptInit onLoadAll

sc_getfield("myField")

Watch a tutorial

This macro is used to assign objects (that represent the field's form) to a javascript variable, allowing the user to access the object properties. This macro will run on control and forms at the javascript events and methods.

EX:

Showing the value of the "client" field of a form cli_js = sc_getfield('cliente'); alert (cli_js.value);

Macro Scope

sc_get_groupby_rule()

Watch a tutorial

This macro is used to retrieve the name of the Group By rule that is being used at this moment.
The return value is a String containing the name of the rule.

Ex:

```
if (sc_get_groupby_rule() == "rule1")
{
[title] = "Sales X Region";
}
```

Macro Scope

Grid application
onScriptInit onNavigate onRecord

sc_get_language

Watch a tutorial

This macro returns the abbreviation of the language being used.

Ex. 1:

$my_language = sc_get_language();

Language	Value
Arabic	ar
Bahasa Melayu	ms
Belarusian	be
Bengali	bn
Bosnian	bs
Bulgarian	bg
Catalan	ca
Chinese Simplified	zh_cn
Chinese Traditional	zh_hk
Croatian	hr
Czech	cz
Danish	da
Dutch	nl
English	en_us
Estonian	et
Finnish	fi
French	fr
Galician	gl
German	de
Greek	el
Gujarati	gu
Hebrew	he
Hindi	hi
Hungarian	hu_hu
Indonesian	id
Italian	it
Japanese	ja
Kannada	kn
Korean	ko
Latvian	lv
Lithuanian	lt
Macedonian	mk
Marathi	mr
Norwegian	no
Polish	pl
Portuguese Brazil	pt_br
Portuguese Portugal	pt_pt
Punjabi	pa
Romanian	ro
Russian	ru
Serbian Cyrillic	sr
Shuar	jiv
Sinhalese	si
Slovak	sk
Slovenian	sl
Spanish	es
Swedish	sv
Telugu	te
Thai	thai
Turkish	tr
Ukrainian	uk
Urdu	ur

sc_get_regional

Watch a tutorial

This macro returns the abbreviation of the regional settings being used.

Ex. 1:

$my_regional = sc_get_regional();

Language	Value
Afrikaans (South Africa)	af_za
Albanian (Albania)	sq_al
Amharic (Ethiopia)	am_et
Arabic (Algeria)	ar_dz
Arabic (Bahrain)	ar_bh
Arabic (Egypt)	ar_eg
Arabic (Iraq)	ar_iq
Arabic (Jordan)	ar_jo
Arabic (Kuwait)	ar_kw
Arabic (Lebanon)	ar_lb
Arabic (Libya)	ar_ly
Arabic (Morocco)	ar_ma
Arabic (Oman)	ar_om
Arabic (Qatar)	ar_qa
Arabic (Saudi Arabia)	ar_sa
Arabic (Syria)	ar_sy
Arabic (Tunisia)	ar_tn
Arabic (United Arab Emirates)	ar_ae
Arabic (Yemen)	ar_ye
Armenian (Armenia)	hy_am
Azerbaijan (Azerbaijan)	az_az
Bahasa Melayu (Malaysia)	ms_my
Basque	eu
Belarusian (Belarus)	be_by
Bengali (Bangladesh)	bn_bd

Macro Scope

Blank calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tre

Bengali (India)	bn_in
Bosnian (Bosnia-Herzegovina)	bs_ba
Breton (France)	br_fr
Bulgarian (Bulgaria)	bg_bg
Catalan (Spain)	ca_es
Chinese (Republic of China)	zh_cn
Chinese (Simplified, Singapore)	zh_sg
Chinese (Traditional, Hong Kong)	zh_hk
Chinese (Traditional, Macau)	zh_mo
Chinese (Traditional, Taiwan)	zh_tw
Croatian (Bosnia and Herzegovina)	hr_ba
Croatian (Croatia)	hr_hr
Czech (Czech Republic)	cs_cz
Danish (Denmark)	da_dk
Dutch (Belgium)	nl_be
Dutch (Netherlands)	nl_nl
English (Australia)	en_au
English (Belize)	en_bz
English (Botswana)	en_bw
English (Cameroon)	en_cm
English (Canada)	en_ca
English (Caribbean)	en_cb
English (India)	en_in
English (Ireland)	en_ie
English (Jamaica)	en_jm
English (Malaysia)	en_my
English (Namibia)	en_na
English (New Zealand)	en_nz
English (Nigeria)	en_ng
English (Philippines)	en_ph
English (Singapore)	en_sg
English (South Africa)	en_za
English (Trinidad and Tobago)	en_tt
English (Uganda)	en_ug
English (United Kingdom)	en_gb
English (United States)	en_us
English (Zambia)	en_zm
English (Zimbabwe)	en_zw
Estonian (Estonia)	et_ee
Faroese (Faroe Islands)	fo_fo
Finnish (Finland)	fi_fi
French (Belgium)	fr_be
French (Cameroon)	fr_cm
French (Canada)	fr_ca
French (France)	fr_fr
French (Ivory Coast)	fr_ci
French (Luxembourg)	fr_lu
French (Monaco)	fr_mc
French (Switzerland)	fr_ch
Frisian, Western (Netherlands)	fy_nl
Galician (Spain)	gl_es
Georgian (Georgia)	ka_ge
German (Austria)	de_at
German (Germany)	de_de
German (Lichtenstein)	de_li
German (Luxembourg)	de_lu
German (Switzerland)	de_ch
Greek (Greece)	el_gr
Gujarati (India)	gu_in
Hausa (Ghana)	ha_gh
Hebrew (Israel)	he_il
Hindi (India)	hi_in
Hungarian (Hungary)	hu_hu
Icelandic (Iceland)	is_is
Indonesian (Indonesia)	id_id
Irish (Ireland)	ga_ie
Italian (Italy)	it_it
Italian (Switzerland)	it_ch
Japanese (Japan)	ja_jp
Kalaallisut (Kalaallit Nunaat)	kl_gl
Kannada (India)	kn_in
Kazakh (Kazakhstan)	kk_kz
Khmer (Cambodia)	km_kh
Kirghiz (Kyrgyzstan)	ky_kg
Korean (Korea)	ko_kr

Lao (Laos)	lo_la
Latvian (Latvia)	lv_lv
Lithuanian (Lithuania)	lt_lt
Luxembourgish (Luxembourg)	lb_lu
Macedonian (Republic of Macedonia)	mk_mk
Malayalam (India)	ml_in
Marathi (India)	mr_in
Mongolian (Mongolia)	mn_mn
Nepali (Nepal)	ne_np
Northern Sotho (South Africa)	nso_z
Norwegian (Norway)	no_no
Norwegian Bokmal (Norway)	nb_no
Norwegian Nynorsk (Norway)	nn_no
Occitan (France)	oc_fr
Oriya (India)	or_in
Persian (Iran)	fa_ir
Polish (Poland)	pl_pl
Portuguese (Angola)	pt_ao
Portuguese (Brazil)	pt_br
Portuguese (Portugal)	pt_pt
Punjabi (India)	pa_in
Romanian (Moldova)	ro_md
Romanian (Romania)	ro_ro
Russian (Belarus)	ru_by
Russian (Kazakhstan)	ru_kz
Russian (Russia)	ru_ru
Serbian Cyrillic (Serbia and Montenegro)	sr_yu
Serbian Latin (Serbia and Montenegro)	sh_yu
Slovak (Slovak Republic)	sk_sk
Slovenian (Slovenia)	sl_si
Sotho (South Africa)	st_za
Spanish (Argentina)	es_ar
Spanish (Bolivia)	es_bo
Spanish (Chile)	es_cl
Spanish (Colombia)	es_co
Spanish (Costa Rica)	es_cr
Spanish (Dominican Republic)	es_do
Spanish (Ecuador)	es_ec
Spanish (El Salvador)	es_sv
Spanish (Guatemala)	es_gt
Spanish (Honduras)	es_hn
Spanish (Mexico)	es_mx
Spanish (Nicaragua)	es_ni
Spanish (Panama)	es_pa
Spanish (Paraguay)	es_py
Spanish (Peru)	es_pe
Spanish (Puerto Rico)	es_pr
Spanish (Spain)	es_es
Spanish (United States)	es_us
Spanish (Uruguay)	es_uy
Spanish (Venezuela)	es_ve
Sri Lanka (Sinhalese)	si_si
Swahili (Tanzania)	sw_tz
Swedish (Finland)	sv_fi
Swedish (Sweden)	sv_se
Tajik (Tajikistan)	tg_tj
Tamil (India)	ta_in
Telugu (India)	te_in
Thai (Thailand)	th_th
Tswana (South Africa)	tn_za
Turkish (Turkey)	tr_tr
Ukrainian (Ukraine)	uk_ua
Urdu (Pakistan)	ur_pk
Uzbek (Uzbekistan)	uz_uz
Vietnamese (Vietnam)	vi_vn
Welsh (United Kingdom)	cy_gb
Xhosa (South Africa)	xh_za
Zulu (South Africa)	zu_za

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tre

onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onFooter onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onA onE onL

sc_get_theme

Watch a tutorial

This macro returns the application theme used.

Ex. 1:

$my_theme = sc_get_theme();

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tre
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onFooter onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onA onE onL

sc_get_wizard_step

Watch a tutorial

This macro works in the form application only when it is configured to work as a step form. With this macro it is possible to identify the current stage that the end user is in and thus perform any necessary validation.

For the macro to work, the form must be configured with at least two steps or more. It has no parameters.

Example: Validation of unfilled fields.

```
if (sc_get_wizard_step() == 1) {

   if ( empty({contactname}) ) {

      sc_error_message("Fill the field Contactname");

   }
}
```

Macro Scope

Form application
onValidate

sc_groupby_label("My_Field")

Watch a tutorial

This Macro dynamically modifies the field label displayed in the group by lines.

Ex. 1:
sc_groupby_label('state') = "state grouping";

Ex. 2:
sc_groupby_label('balance') = "balance total";

Macro Scope

Grid application
onGroupBy

sc_head_hide()

Watch a tutorial

This macro is intended to hide the application header.

With no mandatory parameter, the macro is available in the following applications:

- Form
- Control
- Graphic
- Grid

In the Grid application, in addition to its standard use (without parameters), it is possible to individually define the module where the display of the header will be inhibited.

Below is the list of **unique grid parameters**

- **grid -** Hide only the header of the grid module
- **res** or **sum -** Hide only the summary module header
- **det -** Hides the detail module header

Remembering that the parameters are not mandatory, in the case of the grid application, if no parameter is informed, the header display will be inhibited in all modules.

Examples

Eg. no parameter - Default use of the macro, without passing a parameter, inhibits the display of the header in the Form, Control, Chart and in all modules of the Grid.

```
sc_foot_hide();
```

Below are examples of using the macro with parameter, **exclusively for use in the grid**

Eg. using the 'grid' parameter - Inhibits the header view only in the grid application grid module.

```
sc_foot_hide('grid');
```

Eg. using the 'sum' parameter - Inhibits the header view only in the grid application summary module.

```
sc_foot_hide('res'); or sc_foot_hide('sum');
```

Eg. using the 'det' parameter - Inhibits the header view only in the grid application detail module.

```
sc_foot_hide('det');
```

Eg. using more than one parameter - Inhibits the visualization of the header in the grid application and detail module.

In this case the parameters must be separated by a comma.

```
sc_foot_hide('grid,det');
```

Macro Scope

chart application	Grid application	Form application	Control Form
onScriptInit	onScriptInit	onScriptInit	onScriptInit

sc_hide_groupby_rule('group1', 'grop2', 'group3')

Watch a tutorial

This macro is used to disable Group By rules in execution time.

The Group By rules ID should be informed as parameters in the macro.

This macro will only work if there are at least two group by, so using the macro, one group by will be disabled and other remains showing.

Ex:

```
if( [usr_login_group] == 'seller' ){

sc_hide_groupby_rule('rule_1', 'rule_2');

}
```

Macro Scope

Grid application
onApplicationInit

sc_image(Image01.jpg)

This macro is used to copy images in some events or Javascript commands to load images to an application.

All images used in the applications are automatically copied to each application.

Ex. 1:
sc_image (img1.gif, img2.jpg);

Note: *The images must exist in the development environment images directory (../devel/conf/sys/img/img/).*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu	Re ap
onExecute	onApplicationInit onScriptInit onLoad onValidate	onApplicationInit onScriptInit	onFooter onScriptInit	onApplicationInit onScriptInit	onFilterInit	onApplicationInit onClick onScriptInit onLoad onLoadRecord onValidate	onApplicationInit onScriptInit onLoadAll onValidateFailure onValidateSuccess	onApplicationInit onLoad	onApplicationInit onLoad	onLoad	onA onS

sc_include("File", "Source")

Watch a tutorial

The Scriptcase allows the user to catalog his PHP scripts to use in various applications.

To catalog the scripts go to the ScriptCase main menu at "**Tools > Library**" and use the "upload" option existing in the interface.

When a script is cataloged, the user has the option of the context choice where it will be stored, global level (Scriptcase), group level or user level.
- **Public:** All the projects will have access.
- **Project:** Only the project users (the ones that will be logged at that time) will have access.
- **User:** Only the user (the ones that will be logged at that time) will have access.

The **sc_include** macro, the user must inform the **script name** that will be included and the **source**.
- **Script Name:** Script name to be included.

The Source context that the script was saved, that could be:
- **Sys or Blank:** Scriptcase global level.
- **Prj:** Group level.
- **Usr:** User level.

Ex. 1: Including a catalogued script to global level.
sc_include(my_script.php);

Ex. 2: Including a catalogued script to the group level.
sc_include(my_script.php, grp);

Ex. 3: Including a catalogued script to the user level.
sc_include(my_script.php, usr);

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tre me
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onHeader onScriptInit	onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onE onL

sc_include_lib("Lib1", "Lib2", ...)

Watch a tutorial

This macro is used to dynamically select the application libraries.

PHP Libraries	JavaScript Libraries
tcpdf	Jquery
fpdf_html	jquery_thickbox
excel	jquery_blockui
rtf	tiny_mce
phpspreadsheet	jquery_touch_punch
-	fusioncharts
-	jsmind
-	jkanban
-	orgchart

Ex. 1:

```
sc_include_lib ("tcpdf");

$pdf=new TCPDF();
$pdf->AddPage();
$pdf->SetFont('Helvetica','B',16);
$pdf->Cell(40,10,'Hello World!');
$pdf->Output();
```

Ex. 2:

```
?>
sc_include_lib ("jsmind");
< ?php
```

Macro Scope

Blank application	Grid application
onExecute	onScriptInit

sc_include_library("Target", "Library Name", "File", "include_once", "Require")

Watch a tutorial

This macro includes a **PHP** file from a library in the application. You must have a library created in Scriptcase to make use of this macro. To create a library, go to "Tools -> Libraries".

Parameter	Description
Target	Tells you what the scope of the library. It can be "sys" for libraries of *Public* scope or "prj" for libraries of the *Project* scope.
Library Name	Name given to the library at the time of creation.
File	The absolute path within the library.
include_once (optional)	Make sure that the file will only be included once. If not informed, this value is set as "true".
Require (optional)	If the file or library does not exist, the application to be executed and returns an error. If not informed, the value is set as "true".

Ex: Including a file from a library:
sc_include_library("prj", "phpqrcode", "qrlib.php", true, true);

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tre me
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onScriptInit onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onE onL

sc_label("field_name")

Watch a tutorial

This macro allows dynamically changing the label of fields in grid, form, and control applications.

Parameters

The macro has only one mandatory parameter, which receives the name of the field whose label will be changed.
This parameter must be provided in quotes (double or single) or by a variable, as in the syntax examples below.

```
//Providing as a string
sc_label("customerid") = "New label for the field";

//Providing with a variable
$var_field = "field_name";
sc_label($var_field) = "New label for the field";
```

Examples

Ex. 1: User Registration
In this scenario, a single form is used to register employees and customers.

```
if ({user_type} == 'customer') {
   sc_label('field_id') = 'Customer ID';
} else {
   sc_label('field_id') = 'Employee ID';
}
```

Ex. 2: Event Registration
In an event registration form, the field labels can be changed to reflect the type of event (e.g., Conference, Workshop, Seminar).

```
if ({event_type} == 'conference') {
   sc_label('location') = 'Conference Location';
} else if ({event_type} == 'workshop') {
   sc_label('location') = 'Workshop Location';
} else if ({event_type} == 'seminar') {
   sc_label('location') = 'Seminar Location';
}
```

Macro Scope

calendar application	chart application	Grid application	Dashboard	Form application	Control Form	ReportPDF application
onScriptInit onLoad onRefresh	onScriptInit	onClick onScriptInit onNavigate	onScriptInit	ajaxFieldonBlur Onchange OnClick OnFocus onScriptInit onLoad onNavigate onLoadRecord onRefresh	onBlur onChange onClick onFocus onScriptInit onLoadAll	onScriptInit

sc_language

Watch a tutorial

This macro returns the language and regional settings.

Ex. 1:

echo "Language: ".{sc_language};

Language	Value
Arabic	ar
Bahasa Melayu	ms
Belarusian	be
Bengali	bn
Bosnian	bs
Bulgarian	bg
Catalan	ca
Chinese Simplified	zh_cn
Chinese Traditional	zh_hk
Croatian	hr
Czech	cz
Danish	da
Dutch	nl
English	en_us
Estonian	et
Finnish	fi
French	fr
Galician	gl
German	de
Greek	el
Gujarati	gu
Hebrew	he
Hindi	hi
Hungarian	hu_hu
Indonesian	id
Italian	it
Japanese	ja
Kannada	kn
Korean	ko
Latvian	lv
Lithuanian	lt
Macedonian	mk
Marathi	mr
Norwegian	no
Polish	pl
Portuguese Brazil	pt_br
Portuguese Portugal	pt_pt
Punjabi	pa
Romanian	ro
Russian	ru
Serbian Cyrillic	sr
Shuar	jiv
Sinhalese	si
Slovak	sk
Slovenian	sl
Spanish	es
Swedish	sv
Telugu	te
Thai	thai
Turkish	tr
Ukrainian	uk
Urdu	ur

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	onApplicationInit onCalendarApplicationInit onCalendarScriptInit onScriptInit	onApplicationInit onScriptInit	onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit onValidate onValidateSuccess	onApplicationInit onLoad	onApplicationIn onLoad

sc_ldap_groups

The sc_ldap_groups() macro returns **all existing groups** in **Active Directory (AD)** used in the sc_ldap_login macro.

Syntax

```
sc_ldap_groups();
```

Expected Return

```
Array (
  [N] => Array
      (
         [uniquemember] => Array
            (
               [count] => 4
               [0] => uid=einstein,dc=example,dc=com
               [1] => uid=galieleo,dc=example,dc=com
               [2] => uid=tesla,dc=example,dc=com
               [3] => uid=newton,dc=example,dc=com
            )

         [0] => uniquemember
         [ou] => Array
            (
               [count] => 1
               [0] => scientists
            )

         [1] => ou
         [cn] => Array
            (
               [count] => 1
               [0] => Scientists
            )

         [2] => cn
         [objectclass] => Array
            (
               [count] => 2
               [0] => groupOfUniqueNames
               [1] => top
            )

         [3] => objectclass
         [count] => 4
         [dn] => ou=scientists,dc=example,dc=com
      )
)
```

Macro Scope

calendar application	Form application	Control Form
onApplicationInit onValidate	onApplicationInit onValidate	onApplicationInit

sc_ldap_login($server, $version, $user, $password, $dn, $group, $port, $library)

The sc_ldap_login macro is responsible for establishing the connection with the LDAP (Lightweight Directory Access Protocol) server, enabling connections to **Active Directory (AD)**. It also defines the behavior of other LDAP macros through the $library parameter, which specifies the library to be used (LdapRecord or adLdap).

Syntax

```
sc_ldap_login($server, $version = '', $user = '', $password = '', $dn = '', $group = '', $port, $library);
```

Parameters

Parameter	Description
$server	IP address or hostname of the LDAP server.
$version	LDAP protocol version to be used (example: 3). If left empty, the server's default version will be used.
$user	Username for LDAP authentication. It must be in the format expected by the server, such as cn=admin,dc=company,dc=com or user@company.com.
$password	Password of the user used for LDAP login.
$dn	Distinguished Name (DN) that defines the search point for users. Example: "dc=company,dc=com".
$group	Optionally, a group can be specified to check the user's association.
$port	Port used to connect to the LDAP server (389 for standard connections and 636 for secure SSL connections).
$library	This parameter defines which library will be used for the **LDAP** connection. • **If the value 2 is provided**, the macro will use the **new library** (LdapRecord), available from **version 9.12** of Scriptcase. • **If the parameter is not provided**, the macro will continue using the **legacy library** (adLdap). This configuration allows older projects to continue functioning without modifications, while new implementations can benefit from the enhanced compatibility and security of **LdapRecord**.

Example Usage

```
$server = "ldap.myserver.com";
$version = "3";
$user = "cn=admin,dc=company,dc=com";
$password = "password123";
$dn = "dc=company,dc=com";
$group = "";
$port = 389;
$library = 2; // Or leave empty to use the legacy "adLdap" library

sc_ldap_login($server, $version, $user, $password, $dn, $group, $port, $library);
```

Expected Return

```
Array
(
   [0] => Array
      (
         [sn] => Array
            (
               [count] => 1
               [0] => Newton
            )

         [0] => sn
         [objectclass] => Array
            (
               [count] => 4
               [0] => inetOrgPerson
               [1] => organizationalPerson
               [2] => person
               [3] => top
            )

         [1] => objectclass
         [uid] => Array
            (
               [count] => 1
               [0] => newton
            )

         [2] => uid
         [mail] => Array
            (
               [count] => 1
               [0] => newton@ldap.forumsys.com
            )

         [3] => mail
         [cn] => Array
            (
               [count] => 1
               [0] => Isaac Newton
            )

         [4] => cn
         [count] => 5
         [dn] => uid=newton,dc=example,dc=com
      )
)
```

Macro Scope

calendar application	Form application	Control Form
onApplicationInit onValidate	onApplicationInit onValidate	onApplicationInit onValidate

sc_ldap_logout()

This macro is used to release the connection after using the macro sc_ldap_login.

Ex1:

sc_ldap_logout();

Macro Scope

calendar application	Form application	Control Form
onApplicationInit onValidate	onApplicationInit onValidate	onApplicationInit onValidate

sc_ldap_search($filter = 'all', $attributes = array())

This macro is used to perform searches in the LDAP.

Parameter	Description
$filter	'all' or 'member': 'all' returns all information that the logged user can see. 'member' returns the information about that member
$attributes	Array of LDAP filter options.

Ex:

$result = sc_ldap_search($filter, array('mail', 'displayname'));

Expected Return

```
Array (
  [N] => Array (
         [objectclass] => Array
            (
               [count] => 4
               [0] => inetOrgPerson
               [1] => organizationalPerson
               [2] => person
               [3] => top
            )

         [0] => objectclass
         [cn] => Array
            (
               [count] => 1
               [0] => Albert Einstein
            )

         [1] => cn
         [sn] => Array
            (
               [count] => 1
               [0] => Einstein
            )

         [2] => sn
         [uid] => Array
            (
               [count] => 1
               [0] => einstein
            )

         [3] => uid
         [mail] => Array
            (
               [count] => 1
               [0] => einstein@ldap.forumsys.com
            )

         [4] => mail
         [telephonenumber] => Array
            (
               [count] => 1
               [0] => 314-159-2653
            )

         [5] => telephonenumber
         [count] => 6
         [dn] => uid=einstein,dc=example,dc=com
      )
)
```

Macro Scope

calendar application	Form application	Control Form
onApplicationInit onValidate	onApplicationInit onValidate	onApplicationInit onValidate

sc_ldap_users($filter = 'all', $attributes = array())

The sc_ldap_users() macro is used to retrieve information about users stored on an **LDAP (Lightweight Directory Access Protocol)** server.

With this function, it is possible to retrieve **all attributes visible to the authenticated user** or filter only those they are part of, allowing for **efficient user and permission management**.

Parameters

Parameters	Description
filter	Defines the scope of the search in LDAP. It can be: "all" – Returns **all** information accessible to the authenticated user. "member" – Returns **only** the attributes of the specified users.
attributes	An **array** containing the desired attributes for the query. If omitted, the function returns all attributes available to the authenticated user.

Syntax

```
$users = sc_ldap_users( $filter = 'all', $attributes = array());
```

Expected Return

```
Array
(
   [0] => Array
      (
         [sn] => Array
            (
               [count] => 1
               [0] => Newton
            )

         [0] => sn
         [objectclass] => Array
            (
               [count] => 4
               [0] => inetOrgPerson
               [1] => organizationalPerson
               [2] => person
               [3] => top
            )

         [1] => objectclass
         [uid] => Array
            (
               [count] => 1
               [0] => newton
            )

         [2] => uid
         [mail] => Array
            (
               [count] => 1
               [0] => newton@ldap.forumsys.com
            )

         [3] => mail
         [cn] => Array
            (
               [count] => 1
               [0] => Isaac Newton
            )

         [4] => cn
         [count] => 5
         [dn] => uid=newton,dc=example,dc=com
      )
)
```

Macro Scope

calendar application	Form application	Control Form
onApplicationInit onValidate	onApplicationInit onValidate	onApplicationInit onValidate

sc_link(Column, Application, Parameters, "Hint", "Target", Height, Width)

Watch a tutorial

Allows you to dynamically create a field connection between two applications.

Note: This macro is only available in Query.

Below is the list of macro parameters, which must be informed separated by commas.

Parameter	Value	Description

Column	Accepts only string with or without quote protection. **For example:** sc_link(**customerid**, form_customer, custom={customerid}, "Customer Data", "_self"); Accepted ways of passing the parameter: • customerid • 'customerid' • "customerid"	**Mandatory parameter**, where the name of the field that will be transformed into a link to the other application must be informed. **Note**: If the informed field already has a field link, the link defined in the macro will overwrite the interface link.
Application	Accepts strings with or without the protection of quotes, in addition to global or local variables. **For example:** $var_customer = 'form_customer'; sc_link(customerid, **$var_customer**, custom={customerid}, "Customer Data", "_self"); Accepted ways of passing the parameter: • form_customer • 'form_customer' • "form_customer" • $var_customer • [glob_customer]	**Required parameter**, defines the target application of the call. In this parameter it is possible to use variables making it possible to change the target application, depending on your business rule.
Parameters	This parameter defines the value of the variable that will be available from the target application as a global variable. **For example:** sc_link(customerid, form_customer, **custom={customerid}**, "Customer Data", "_self"); Accepted ways of passing the parameter: • custom = {customerid} • custom = 'ALFKI' • custom = $var_customer • custom - [glob_customer]	This parameter is not mandatory, but its use is necessary for the correct display of data in the target application. When not informed, the target application will display all data in the table. **Retrieving the value in the target** application In the target application of the connection, in the example on the side is the form_customer application, the **custom** must be retrieved as a global variable in the application's SQL. In this example, the following SQL statement has been added to the form (form_customer) customerid = '[custom]' For more details on global variable usage, access our online documentation.
Hint	Accepts string, with or without the protection of quotes, lang and global or local variables. **For example:** sc_link(customerid, form_customer, custom={customerid}, **"Customer Data"**, "_self"); **Obs**: To use **commas** and **apostrophes** in the text, it is necessary to protect with the scape **"\"** Accepted ways of passing the parameter: • "Help text\, with a comma" • 'Customer • Data' Customer Data • {lang_text} • [glob_txt] • $ var_txt	This parameter is not mandatory. Receives a text with some help or remark for the system user, the text is displayed whenever the mouse cursor passes over the link. It is possible to use langs for the definition of messages, allowing the translation of your system. **Obs**: To use **commas** and **apostrophes** in the text it is necessary to protect with the scape **"\"**

Target	This parameter accepts only string. Available values are listed below. **For example:** sc_link(customerid, form_customer, custom={customerid}, "Customer Data", **"modal"**, 850, 600); **Note:** If omitted, the value "_self" will be assigned. Accepted ways of passing the parameter: • "_self" • "_blank" • "iframeT" • "iframeL" • "iframeR" • "iframeB" • "modal"	Defines the way to open the target application. • _self - Runs the target application on the same page, overlapping the query application; • _blank - Runs the target application on another page (pop-up); • iframeT - Displays the target application in an iframe above the query (Top); • iframeL - The target application will be displayed in an iframe to the left of the query (Left) • iframeR - The target application will be displayed in an iframe to the right of the query (Right); • iframeB - The target application will be displayed in an iframe below the query (Bot); • modal - Opens a new window and blocks any interaction in the main window; When using any of the **iframe** or **modal** to open the application, it is possible to define the **height** and **width** by adding the values in the optional parameters below.
Height	This parameter accepts only numerical values and its unit of measurement is pixel. **For example:** sc_link(customerid, form_customer, custom={customerid}, "Customer Data", "modal", **850**, 600);	**Non-mandatory parameter**, available only when using the **Iframe** and **modal** parameter **Target** Defines the height of the iframe or modal where the target application will be displayed. **Note**: If omitted, the default value will be 440
Width	This parameter accepts only numerical values and its unit of measurement is pixel. **For example:** sc_link(customerid, form_customer, custom={customerid}, "Customer Data", "iframeB", 850, **600**);	**Non-mandatory parameter**, available only when using the **Iframe** and **modal** parameter **Target** Defines the width of the iframe or modal where the target application will be displayed. **Note:** If omitted, the default value will be 630

Examples

Ex. 1: Creates a link to the costumer.php application, on the Id field, passing the global variable [global_costumer] as and the product field (local variable) {Id} to the costumer.php application:

```
sc_link(Id, costumer.php, product=[global_costumer]; Id={Id}, "Client Data", "_blank");
```

Ex. 2: Creating a dynamic link to another application depeding on the field valor {costumer_type} and opening it in a modal.

```
if ( {costumer_type} == "F") {

   sc_link (costumer_type, type_f.php, product=[global_costumer]; Id={Id}," Personal Data ", "modal", 500, 700);

} else {

   sc_link (costumer_type, type_a.php, product=[global_costumer]; Id={Id}," Company Data ", "iframeR");
}
```

Ex. 3: Creating a link to a URL. A redirect to the given URL will be performed.

```
sc_link (my_field, http://www.scriptcase.com, ,"hint of the link", "_blank");
```

Macro Scope

Grid application
onRecord

sc_log_add("action", "description")

Watch a tutorial

This macro is intended to add a record to the log table, thereby making the message and/or the name of the action customizable to the developer. The macro parameters can use PHP variables.

Ex:

sc_log_add ("access", "Application has been accessed by the user trial");

Macro Scope

Blank application	calendar application	Grid application	Dashboard	Form application
onExecute	onValidate	onScriptInit	onWidgetLoad	onValidate

sc_log_split({description})

Watch a tutorial

This macro is intended to provide access to the information written in the last insertion in the log table, allowing the manipulation of the data before and after the record update returning it in an array mode. The information is stored in the "description" field of the log table in the string format with delimiters.

Ex:

$arr_description = sc_log_split({description});

Input: -> keys fields Array ([clientid] => ANTON) [fields] => Array ([old] => Array ([CEP] => 50710500 [Address] => Rua Desembargador Joao Paes 657 [City] => Farol [State] => RS) [new] => Array ([CEP] => 53230630 [Address] => Avenida Presidente Kennedy, 1001 [City] => Peixinhos [State] => PE)))

Macro Scope

Blank application	calendar application	Grid application	Dashboard	Form application
onExecute	onLoad	onScriptInit	onWidgetLoad	onLoad

sc_lookup(Dataset, "SQL Command", "Connection")

Watch a tutorial

This macro allows the user to **execute SQL commands** and returns the result to the "dataset" variable. The "dataset" structure **is an array** (line/column).

The "**connection**" parameter is optional. Use when the command is executed in a database different from that specified for the application.
Note: The connection parameter does not accept variable. You must enter the name of the connection that the SQL command will execute.

Ex. 1:
sc_lookup(dataset, "select customer_id, customer_name, credit_limit from customers");

To have access to the first line (Dataset), use :
{customer_id} = {dataset[0][0]};
{customer_name} = {dataset[0][1]};
{credit_limit } = {dataset[0][2]};

To have access to the second line (Dataset), use :
{customer_id} = {dataset[1][0]};
{customer_name} = {dataset[1][1]};
{credit_limit} = {dataset[1][2]};

If occurs **error in the execution of the SQL command** , the variable attributed to the dataset will return as "**false**" and the error message will be available in the "**dataset_error**" variable. It is also important to verify the select returned data, to prevent access to non-existent variables, once the output array only will be created if the select command returns data.

Ex. 2:

```
sc_lookup(my_data, "select customer_id, customer_name, credit_limit from customers");
if ({my_data} === false)
{
echo "Access error. Message=". {my_data_error} ;
}
elseif (empty({my_data}))
{
echo "Select command didn't return data";
}
else
{
{customer_id} = {my_data[0][0]};
{customer_name} = {my_data[0][1]};
{credit_limit} = {my_data[0][2]};
}
```

Ex. 3: The SQL command also can be composed of application fields (local variables) or of global variables:
sc_lookup(dataset, "select order_value from orders where clientid = '{customer_id} ' and salesman_id = [var_glo_salesman]");

Note: *The command must always be finished with a semicolon ";".*

Note2: *For a big result returned in the dataset we recommend the use of the macro sc_select instead of this one.*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree me

onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarScriptInit onScriptInit onLoad onRefresh onValidate onValidateSuccess	onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateSuccess	onBlur onChange onClick onFocus onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onApplica onExecut onLoad

sc_lookup_field(Dataset, "SQL Command", "connection_name")

This macro allows the developer to **execute a SELECT command at a time** and have access to the data returned through the dataset parameter, in array format: {dataset[line_number][field_name]}.
To run commands that return a large amount of data, we recommend using sc_select macro for better performance.

Unlike the macro sc_lookup, where we inform the index {dataset[0][0]}, in the macro sc_lookup_field we must inform the name of the column that we want to retrieve the data {dataset[0][field_name]}

> The way of assembling the return array, where we must inform the column name, makes this macro incompatible with the following databases: **MSSQL Server**, **Access** and **DB2**. In these cases, we recommend using the sc_lookup macro.

See below for details on the macro parameters.

Parameter	Description	Examples

dataset	This parameter is mandatory and **must be informed without spaces or special characters and does not accept variables.**
	
	It defines the name of the variable that will receive the return of the SELECT command executed by the macro.
	
	In case of error in the execution of the SQL command, this parameter will return false and the error message will be available in the variable **"dataset_erro".**
	
	Below is a simple example of validation and error message display.
	`if ( {meus_dados} === false ) {`
	`echo "Access error. Message = " . {meus_dados_erro};`
	`}`
		`sc_lookup_field(`**"my_dataset"**`, "select field_01, field_02, field_03 from tb_name");`
		
		When executing the command **print_r({my_dataset});** we will get the following return:
		`Array (`
		`[0] => Array (`
		`[field_01] => 1`
		`[field_02] => OFFICE EQUIPMENT`
		`)`
		`[1] => Array (`
		`[field_01] => 2`
		`[field_02] => FIXED TELEPHONY`
		`)`
		`)`
		
		In the example above, the command returned **two record lines** and in each line two columns (field_01 and field_02).
		
		The recovery of the value should take place as follows.
		
		{my_dataset[line_number][colunm_name]}

SQL Command	**This parameter is mandatory** and can be defined by informing the command directly in the macro protected with quotes (double or single) or using a variable (local or global).	`// Command directly in the Macro`
		`sc_lookup_field("retorno_select", "select field_01, field_02, field_03 from tb_name");`
		
		`// Sample with variable`
		`$comando_select = "select field_01, field_02, field_03 from tb_name"`
		`sc_lookup_field("retorno_select", $comando_select);`

Connection	The "connection" **parameter is optional**, and must be informed if the command is executed in a database different from the one specified for the application. Furthermore, this parameter does not accept the use of variables, being necessary to type the name of the connection for the macro to run correctly.	`// Sample of using the connection parameter` `sc_lookup_field("retorno_select", "select field_01, field_02, field_03 from tb_name", "nome_conexao");`

Examples

Example 1: Access multiple lines

```
sc_lookup_field(my_dataset, "select field_01, field_02, field_03 from tb_name");

//To access the first line (dataset), we must inform:
{field_01} = {my_dataset[0]['field_01']};
{field_02} = {my_dataset[0]['field_02']};
{field_03} = {my_dataset[0]['field_03']};

/To access the second line (dataset), we must inform:
{field_01} = {my_dataset[1]['field_01']};
{field_02} = {my_dataset[1]['field_02']};
{field_03} = {my_dataset[1]['field_03']};
```

Example 2: With data validation:

```
$comando_sql = "select field_01, field_02, field_03 from tb_name";

sc_lookup_field(my_dataset, $comando_sql);

if ( {my_dataset} === false ) {

   echo "Access error. Message = " . {my_dataset_error};

} elseif ( empty({my_dataset}) ) {

   echo "select command did not return data ";

} else {

   {clienteid} = {my_dataset[0]['field_01']};
   {nomecliente} = {my_dataset[0]['field_02']};
   {limitecred} = {my_dataset[0]['field_03']};

}
```

Example 3: Using the connection parameter

```
$comando_sql = "select field_01, field_02 from tb_name where field_03 = '{field_03} ' and field_02 = [var_global]"

sc_lookup_field(my_dataset, $comando_sql, "conn_name");
```

Example 4: Using local and global variables in the macro command

```
$comando_sql = "select field_01, field_02 from tb_name where field_03 = '{field_03} ' and field_02 = [var_global]"

sc_lookup_field(my_dataset, $comando_sql);
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Tree menu	ReportPDF application
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarScriptInit onScriptInit onLoad onRefresh onValidate onValidateSuccess	onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateSuccess	onBlur onChange onClick onFocus onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onFooter onHeader onScriptInit onRecord

sc_mail_send(SMTP, Usr, Pw, From, To, Subject, Message, Mens_Type, Copies, Copies_Type, Port, Connection_Type, Attachment, SSL)

Watch a tutorial

This macro is used to send emails.

Parameter	Description
SMTP	SMTP server name or IP address. (String or Variable that contains the server name)
Usr	SMTP username. (String or Variable that contains the username)
Pw	SMTP password. (String or Variable that contains the password)
From	From email. (String or Variable that contains the email)
To	List of the emails that will receive the message, it could be a string or variable that contains one or more emails separated by ";" or one variable that contains one array of emails.
Subject	Message subject. (String or Variable that contains the subject)
Message	Message body. (String or Variable that contains the message)
Mens_Type	Message format: (T) Text or (H) HTML.
Copies	List of the emails that will receive the message, it could be a string or variable that contains one or more emails separated by ";" or one variable that contains one array of emails.
Copies_Type	Type copies: BCC (Hidden copies) or CCC (Regular copies)**.**
Port	The port used by your e-mail server. Use the port **465** for the security type SSL, and the port **587** for the security type TLS or the port **25** for no security. If not informed, Scriptcase will set the default port: **25**
Connection_Type	Indicating a secure connection. Use **S** for SSL, **T** for TLS or **N** for non-secure connections. If not informed, Scriptcase will set the default value:**N**.
Attachment	Absolute path of the file that will be attached in the email
SSL	Optional parameter to add SSL settings. If you need to add more than one setting, use ";" for separate them. Ex: $ssl = 'allow_self_signed=false;verify_peer=false';
Reply_to	It is the email address the reply message is sent to, when you want the reply to go to an email address other than the From address.

Note: To use safe connection **its required** to have the PHP "**php_openssl**" extension enable.

Example 1: Sending e-mail with reply_to parameter

To use the replay_to parameter, you must include all the other parameters before it, just inform them empty.

```
sc_mail_send('smtp.meuserver.com.br', 'usr', 'pw', 'from@netmake.com.br', 'to@netmake.com.br', 'test sending
of e-mail', 'message', 'H', 'abc@cop.com.br;zxy@cop.com.br', 'CCC', '587', '', 'c:/teste/arq .txt','','reply@netmake.com.br');
```

Example 2: Using variables and defining the port.

```
sc_mail_send([glo_smtp], [usuar], [glo_pw], {from}, {to}, {subject}, {message}, 'T', [glo_copias], 'CCC', '419', 'S', {attached});
```

Example 3: No user and password.

The amount of emails sent will be stored at the special variable sc_mail_count.
If any error happens when the Scriptcase try to send the email, the variable sc_mail_ok will receive "**false**" and the error message will be stored on the variable sc_mail_erro.

```
sc_mail_send([glo_smtp], '', '', 'from.netmake.com.br', {to}, {subject}, {message}, 'H', '', '', '', '', [glo_att]);
```

Example 4: Email validation with {sc_mail_ok}

```
sc_mail_send([glo_smtp], '', '', 'from@netmake.com.br', {to}, {subject}, {message}, 'H');

if ({sc_mail_ok}) {
```

```
 echo "sent {sc_mail_count} e-mail(s) with success!!";

} else {

 sc_error_message({sc_mail_erro});

}
```

Example 5: Sending attached files

```
//Defining macro parameters in variables
$mail_smtp_server   = 'smtp.gmail.com';  // SMTP Server
$mail_smtp_user     = 'doc@gmail.com.br';  // SMTP access user
$mail_smtp_pass     = 'password';          // Password for the SMTP user entered above
$mail_from          = 'doc@gmail.com.br';       // Message origin email
$mail_to            = 'doc@gmail.com.br';         // Email of message recipient
$mail_subject       = 'Subject';             // Email subject
$mail_message       = 'Email body';          // Email body
$mail_format        = 'T';                   // Email body format: T (Text only) or H (For text and HTML)
$mail_copies        = '';                    // Parameter that defines the copied emails.
$mail_tp_copies     = '';                  // Copy type: BCC (For blind copy) or CCC (For normal copy)
$mail_port          = '465';                 // Sending server port
$mail_tp_connection = 'S';            // Use or not of secure connection: Y (Secure connection) or N (Insecure connection)
$mail_attachments   = 'c:/teste/arq.txt' // Directory of the file that will be sent as an attachment

//Macro with variables
sc_mail_send($mail_smtp_server,
     $mail_smtp_user,
          $mail_smtp_pass,
     $mail_from,
     $mail_to,
     $mail_subject,
     $mail_message,
     $mail_format,
     $mail_copies,
     $mail_tp_copies,
     $mail_port,
     $mail_tp_connection,
     $mail_attachments);
```

Example 6: Sending multiple files as attachments

```
//Parâmetros do e-mail
$mail_smtp_server   = 'smtp.gmail.com':   // SMTP Server
$mail_smtp_user     = 'doc@gmail.com.br'; // SMTP access user
$mail_smtp_pass     = 'password';         // Password for the SMTP user entered above
$mail_from          = 'doc@gmail.com.br'; // Message origin email
$mail_to            = 'doc@gmail.com.br'; // Email of message recipient
$mail_subject       = 'Subject';          // Email subject
$mail_message       = 'Email body';       // Email body
$mail_format        = 'T';                // Email body format: T (Text only) or H (For text and HTML)
$mail_copies        = '';                 // Parameter that defines the copied emails.
$mail_tp_copies     = '';                 // Copy type: BCC (For blind copy) or CCC (For normal copy)
$mail_port          = '465';              // Sending server port
$mail_tp_connection = 'S';                // Use or not of secure connection: Y (Secure connection) or N (Insecure connection)
// Array of the file that will be sent as an attachment.
$mail_attachments   = array('C:\Program Files\NetMake\v9-php81\wwwroot\scriptcase\file\img\color.png','C:\Program Files\NetMake\v9-php81\wwwroot\scriptcase\file\img\test.txt');

//Macro with variables
sc_mail_send($mail_smtp_server,
     $mail_smtp_user,
          $mail_smtp_pass,
     $mail_from,
     $mail_to,
     $mail_subject,
     $mail_message,
     $mail_format,
     $mail_copies,
     $mail_tp_copies,
     $mail_port,
     $mail_tp_connection,
     $mail_attachments);
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu
onExecute	onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onHeader onScriptInit onRecord	onScriptInit	onFilterInit onFilterSave onFilterValidate	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onApplicatior onExecute onLoad

sc_make_link(Application, Parameters)

Watch a tutorial

This macro creates a string containing a link to other Scriptcase applications (when writing HTML code).

Parameter	Description
Application	Name of the application to establish the link.
Parameters	Parameters necessary so-called application. In the format: "=" (the attribution sign) . More than one parameter must be separated by ";" (semicolon).

Ex. 1: Creating a string with a link for the datacli.php application, without parameters passage.
$string_link = sc_make_link(datacli.php);

Ex. 2: Creating a string with a link for the application datacli.php, with parameters passage.
$string_link = sc_make_link(datacli, parm1={company};parm2=[glo_office]);

Macro Scope

Blank application	chart application	Grid application	Control Form
onExecute	onFooter onHeader onScriptInit	onScriptInit onNavigate onRecord	onValidate onValidateSuccess

sc_master_value("Object", "Value")

Watch a tutorial

Update data in the **master** application, in real-time, according to changes made in a given field of the **detail** application.

Parameter	Description
Object	Object name to be updated in the **master** Application. It's not necessary the {} in a field name.
Value	Object value. The value can be a field or a variable.

Ex. 1:

```
sc_lookup(result,"SELECT SUM(Total) FROM adm_order_itens WHERE OrderID = '{OrderID}'");

//Using sc_format_num() to format the value that will be updated in the master application
$value = sc_format_num({result[0][0]}, ',', '.', 2, 'S', 1, '');
sc_master_value('OrderPrice', $value);
```

** If the value, that will be sent to the master, be numeric, you have to use the **sc_format_num()** to format it. To see the macro documentation, click here.*

Macro Scope

calendar application	Form application	Control Form
onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure	onScriptInit onLoadAll onRefresh

sc_menu_delete(Id_Item1)

Watch a tutorial

This macro removes menu items passed as parameters. It can be used in menu applications or tree menu.

Observe that the parameters used in this macro are the "ID" of the items.

If a menu item is removed and it has "nodes", they will be hidden too.

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_menu_disable(Id_Item1)

Watch a tutorial

This macro deactivates menu items passed as parameters. Can be used in menu applications or tree menus.

The parameters passed are items ï¿½IDï¿½. Menu items are deactivated recursively (when a submenu item is deactivated its ï¿½nodesï¿½ are also deactivated).

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_menu_force_mobile(boolean)

Watch a tutorial

The sc_menu_force_mobile macro forces the display in mobile mode in applications of type Menu (horizontal or TreeMenu).

> This macro is not compatible with the Responsive Menu application.

Parameters

- **true:** Forces the application to run in mobile format, regardless of the device.
- **false:** The application will be displayed according to the type of device, respecting automatic detection.

Examples:

Example 1: Forcing the mobile display of the application.

```
sc_menu_force_mobile(true);
```

Example 2: Returns to the default behavior (automatic detection).

```
sc_menu_force_mobile(false);
```

Macro Scope

Menu application	Tree menu
onLoad	onApplicationInit onLoad

sc_menu_item

Watch a tutorial

This macro is only available for menu applications and it has the objective of giving access to a menu item that was selected, this way the user can take decisions before running the application.

ï¿½

Ex. 1: To clean global variable.

```
if ({sc_menu_item} == "item_1")
{
sc_reset_global([global1], [global2]);
}
```

Ex. 2: If an application requires a global variable another application can initialize it accordingly.

```
if ({sc_menu_item} == "item_5" && (!isset([glo_employ]) || empty([glo_employ])))
{
sc_redir(apl_inf_emp);
}
```

ï¿½

Ex. 2: Use the button on the toolbar menu.

```
if ({sc_menu_item} == "btn_1")
{
sc_apl_conf("form_customer", "start", "new");
}
```

ï¿½

Macro Scope

Menu application	Responsive Menu	Tree menu
onExecute	onExecute	onApplicationInit onExecute

sc_redir('app_name/url', parameter01; parameter02, 'target', 'error', 'modal_height', 'modal_width')

Watch a tutorial

The **sc_redir** macro is used to create dynamic redirects between two applications or between an application and a URL, allowing the passing of parameters and control over how the destination application will be opened using the **target** parameter.

> In form applications, the **sc_redir** macro can be used in events that depend on database updates, such as: **onAfterInsert**, **onAfterUpdate**, **onAfterDelete**, **onBeforeInsert**, **onBeforeUpdate**, and **onBeforeDelete**.
> However, in these events, the **sc_commit_trans()** macro must be used before **sc_redir** to confirm the transaction; otherwise, the redirection will occur without the table changes being applied.

Parameters

Parameter	Description	Examples
app_name/url	Receives the name of the destination application or URL to which the user will be redirected. The values must be enclosed in double or single quotes.	sc_redir(**'app_name'**);
parameter_to_send	Allows passing values to the destination application in the format: parameter_name='value'. You can use a application_field, variable or informa a fixed value. If more than one parameter is passed, use a semicolon (;) as a delimiter. Values should be retrieved as global variables in the destination application.	sc_redir('app_name', **parm1=50**); Two parameters passed: sc_redir('aplx', parm1={clienteid}; parm2="xxx"); In 'aplx', values retrieved as global variables: [parm1] and [parm2].

target	Defines how the destination application/URL will be displayed.
	
	• **_self**: Opens the destination in the same tab (**default if no value is specified**).
	• **_blank**: Opens the destination in a new tab.
	• **_parent**: Opens the destination in the parent window, if the current page is in an iframe.
	• **modal**: Opens the destination in a modal window (**this option is unavailable when using the macro in a Run button**)
		Specifying target without parameters:
		sc_redir('app_name', , **'_blank'**);
		
		Using modal:
		sc_redir('app_name', , **'modal'**, , **'800'**, '600');

error	Defines the behavior of the redirection in case of an error in the application. • **'E'**: Blocks the redirection if an error occurs. • **'F'**: Forces the redirection even if an error occurs.	sc_redir('app_name', , ' ', **'E'**);
modal_height	Defines the height of the modal in pixels. This parameter is mandatory when using modal in the target parameter.	sc_redir('app_name', , 'modal', , **'800'**, '600');
modal_width	Defines the width of the modal in pixels. This parameter is mandatory when using modal in the target parameter.	sc_redir('app_name', , 'modal', , '800', **'600'**);

Usage Examples

Example 1 - Simple redirection without parameters.

```
//Since only the application was specified, it will open in the same tab.
sc_redir('aplx');
```

Example 2 - Redirection with parameter passing.

```
//Four parameters are being sent to the destination application; these values must be retrieved as global variables: [parm1], [parm2], [parm3], [parm4].
if (empty($variable)) { $variable = "documentation"; }
sc_redir('aplx', parm1={clienteid}; parm2="xxx"; parm3=$variable; parm4=10);
```

Example 3 - Using target without parameters.

```
//In this case, since no parameters are being sent to the destination application, the space should be left blank to specify the target.
sc_redir('aplx', , '_blank');
```

Example 4 - Using modal for the target parameter.

```
//It is mandatory to define the width and height when specifying modal in the target parameter.
//Only the values should be provided.
sc_redir('aplx', , "modal", '', '800', '600');
```

Example 5 - Example of use in form database transaction events.

```
//The sc_commit_trans macro must be used before sc_redir to confirm the transaction.
sc_commit_trans();
sc_redir('aplx', parm1={clienteid});
sc_set_app_conf("Application", "Property")
```

Example 6 - Redirection to a URL.

```
if ([glo_usr] == "test")
{
sc_redir(http://www.mypage.com.br);
}
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsi Menu

Watch a tutorial

This macro has the objective to delete all modifications effected by the **sc_apl_conf** macro. The property parameter its optional, if it's passed it will only erase the modification from that specific application.

Ex. 1: Deleting the modifications of the "my_application" application, property "start".
sc_reset_apl_conf("my_application", "start");

Ex. 2: Deleting the modifications of the "my_application" application.
sc_reset_apl_conf("my_application");

Ex. 3: Deleting the modifications of all the applications.
sc_reset_apl_conf();

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsiv Menu
onExecute	onApplicationInit onCalendarApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit onValidate onValidateSuccess	onApplicationInit onLoad	onApplicati onLoad

sc_reset_apl_default

Watch a tutorial

This macro should be used when necessary to remove **sc_apl_default()** macro control over the applications.

Ex1:

sc_reset_apl_default();

Macro Scope

calendar application	chart application	Grid application	Form application	Control Form	ReportPDF application
onScriptInit	onScriptInit	onScriptInit	onScriptInit	onScriptInit	onScriptInit

sc_reset_apl_status

Watch a tutorial

This macro has the objective to clean all security variables, set through the sc_apl_status macro.

Ex. 1:

sc_reset_apl_status();

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsiv Menu
onExecute	onApplicationInit onCalendarApplicationInit onCalendarScriptInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit onValidate onValidateSuccess	onApplicationInit onLoad	onApplicati onLoad

sc_reset_change_connection

Watch a tutorial

This macro has the objective to erase the changes made using the macro "sc_change_connection".

Ex. 1:

sc_reset_change_connection();

Macro Scope

Blank application	calendar application	chart application	Grid application	Form application	Control Form	Menu application	Responsive Menu	Tree menu	ReportPDF application
onExecute	onApplicationInit onCalendarApplicationInit onCalendarScriptInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit onValidate onValidateSuccess	onApplicationInit onLoad	onApplicationInit onLoad	onLoad	onScriptInit

sc_reset_connection_edit

Watch a tutorial

This macro undoes the connection edits made by macro "sc_connection_edit".

Ex. 1:

sc_reset_connection_edit();

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu	Re ap
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarApplicationInit onCalendarScriptInit onScriptInit onLoad onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onScriptInit	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onExecute onLoad	on on on on on

sc_reset_connection_new

Watch a tutorial

This macro undoes the connections made by the macro "sc_connection_new".

sc_reset_connection_new();

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu	Re ap
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarApplicationInit onCalendarScriptInit onScriptInit onLoad onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onScriptInit	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onExecute onLoad	on on on on on

sc_reset_global([Global_Variable1], [Global_Variable2] ...)

Watch a tutorial

This macro is used to delete global variables, stored in the PHP session.

Ex. 1:

sc_reset_global ([Login], [Pass]);

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Respons Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarApplicationInit onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onGroupBy onHeader onScriptInit onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplica onExecut onLoad

sc_reset_menu_delete

Watch a tutorial

This macro its used to restore a menu item, deleted by the macro "sc_menu_delete".

Ex. 1:

sc_reset_menu_delete();

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_reset_menu_disable

Watch a tutorial

This macro has the objective to enable a menu item, disabled by the macro "sc_menu_disable".

Ex. 1:

sc_reset_menu_disable();

Macro Scope

Menu application	Responsive Menu	Tree menu
onApplicationInit onLoad	onApplicationInit onLoad	onApplicationInit onLoad

sc_rollback_trans("Connection")

Watch a tutorial

This macro is used to cancel a transaction set in the database.

The "Connection" parameter is optional, use only if the command is executed in a different data base from the specified to the application.

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onClick onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onExecute onLoad

sc_script_name

Watch a tutorial

This macro is available to the MENU applications to access the application name that was selected in the menu. It allows the user to make decisions before executing the application if a pre-process is required to that application.

Ex. 1: To clean global variables.

```
if ({sc_script_name} == "apl1")
{
sc_reset_global([global1], [global2]);
}
```

Macro Scope

Menu application	Responsive Menu	Tree menu
onExecute	onExecute	onExecute

sc_select(dataset, "SQL Command", "Connection")

This macro executes the SQL commands passed as parameter and access the "dataset" in the command.

Different from sc_lookup macro, this macro doesn't manipulate the dataset (the user is responsible for all the manipulation).

If an error occurs in the sql command execution, the variable attributed to the database returns as "false" and the error message is available in the "dataset_error" variable.

The connection parameter is optional, use only if the command is executed in a data base different from the specified in the application. **In this parameter it is not possible to use variables.**

Ex. 1:

```
sc_select(my_data, "select clientId, clientName, limitecred from costumers");
if ({my_data} === false)
{
echo "Access error. Message =". {my_data_erro};
}
else
{
while (!{my_data}->EOF)
{
{clientName} = {my_data}->fields[1];
{my_data}->MoveNext();
}
{my_data}->Close();
}
```

Ex. 2: The SQL command can passed as application fields (local variables) or of global variables.

```
sc_select(dataset,"select price order from order where clientId = '{clientId}' and cod_Seller =
[var_glo_seller]");
```

Note: *The command must always be finished with semicolon";".*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarScriptInit onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onExecut onLoad

sc_select_field({Field})

Watch a tutorial

This macro modify dynamically the SELECT:
Is possible to modify dynamically the grid fields through the dynamic modification of the original select.

Ex. 1:
A table with the columns year and the months from January to December is used to create a grid using a parameter, displays the years and only a specific month. Passing the parameter as a global variable named "show_month", in the application is found the following structure:

Select Clause: select year, pair_month from table

sc_select_field(pair_month) = [show_month];
sc_label(pair_month) = [show_month];
Modify the select, and the grid column label.

It is possible, during the grid execution time, modify the grid ORDER BY clause.
At least one ORDER BY is required in the original grid SQL command to allow the substitution execution during the grid time.

Ex. 2: Select Clause: select code, name, value from table order by code. To change the order, from code to name, use: sc_select_order("code") = "name";

Note: *This command must be in the context of the event "onInit";*

Macro Scope

chart application	Grid application
onScriptInit	onScriptInit

sc_select_order("Field")

Watch a tutorial

It is possible, during the grid execution time, to modify the grid ORDER BY clause.
At least one ORDER BY is required in the original grid SQL command to allow the substitution during the grid execution time.

Ex. 1: Select Clause: SELECT code, name, price FROM table ORDER BY code
To change the order from code to name, we will have:
sc_select_order("code") = "name";

Ex. 2: Select Clause: SELECT code, name, price, date FROM table ORDER BY code, name, price
To change the order from name to date, we will have:
sc_select_order("name") = "date";

Note: *This command must be in the context of "process before the select".*

Macro Scope

chart application	Grid application
onScriptInit	onHeader onScriptInit

sc_select_where(add)

Watch a tutorial

This is possible, during grid execution time to add a field/condition to the search WHERE clause.

Ex. 1: It adds the content between the quotes (") in the grids select.

```
if (empty({sc_where_current}))
{
sc_select_where(add) = "where campoX > [global]";
}
else
{
sc_select_where(add) = "AND campoX > [global_variable]";
}
```

Note: *This command must be in the context of "process before the select".*

Macro Scope

chart application	Grid application
onScriptInit	onHeader onScriptInit

sc_send_mail_api($arr_settings)

Watch a tutorial

This macro allows dynamic sending of embedded emails with the **Mandrill** and **Amazon SES** APIs. It is also possible to send e-mail using the **SMTP** protocol.

> This macro uses the same mandrill parameters to send e-mail, regardless of the Gateway (Changer, Amazon SES or SMTP) selected. click here and see the list of parameters that can be used.

This macro contains a single parameter "**$arr_settings**", which receives an array with the email sending information.

> For more information on the parameters the mandirll click here

.

Parameter	Description
$arr_settings	Array with information for use of APIs.

Array indices	Description
profile	Name of some API already configured in "Tools> API", as can be seen in Example 1. When informing a profile, the **settings** index should not be informed.
settings	E-mail sending server settings. The information must conform to the API used. Note: When informing the settings, the Profile index is disregarded.
message	Message that will be sent by email. Subject, body, to, etc.

Example 1: Sending e-mail using a pre-configured Profile in "Tools> API".

```
if({api_type} == 'smtp' ){
 $var_config = array(
  'profile' => 'yahoo_example',

  'message' => [
   'html'          => {email_body},
   'text'          => '',
   'to'           => $arr_merge,
   'subject'       => {subject}
  ]
 );
}

sc_send_mail_api($var_config);
```

Example 2: Sending e-mail via SMTP

```
$smtp_server = "smtp.mail.yahoo.com";
$smtp_port = "465";
$smtp_user = "scriptcase.export";
$smtp_password = "scriptcase.export";
$from_email = "scriptcase.export@yahoo.com";
$from_name = "Scriptcase Export";

if ( {api_type} == 'smtp' ){
 $var_config = array(

  'profile' => '',

  'settings' => [
    'gateway'       => 'smtp',
    'smtp_server'   => $smtp_server,
    'smtp_port'     => $smtp_port,
    'smtp_user'     => $smtp_user,
    'smtp_password' => $smtp_password,
    'from_email'    => $from_email,
    'from_name'     => $from_name
  ],

  'message' => [
      'html'          => {email_body},
      'text'          => '',
      'to'           => $arr_merge,
      'subject'       => {subject}
  ]
 );
}

sc_send_mail_api($var_config);
```

Example 3: Sending email using Mandrill API

```
if({api_type} == 'mandrill' ){
 $var_config = array(
  'settings' => [
      'gateway'       => 'mandrill',
      'api_key'       => {api_key},
      'from_email'    => {from_email},
      'from_name'     => {from_name}
  ],

  'message' => [
      'html'          => {email_body},
      'text'          => '',
      'to'           => array($arr_merge),
      'subject'       => {subject}
  ]
 );
}

sc_send_mail_api($var_config);
```

Example 4: Sending email using Amazon SES API

```
if({api_type} == 'ses' ){
 $var_config = array(
  'settings' =>  [
       'gateway' => 'ses',
```

```
        'region' => {ses_region},
        'api_key' => {api_key},
        'api_secret' => {ses_secret},
        'from_email'    => {from_email}
    ],
    'message' => [
        'html'          => {email_body},
        'text'          => '',
        'to'            => array($arr_merge),
        'subject'       => {subject}
    ]
 );
}

sc_send_mail_api($var_config);
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Tree menu	ReportPDF application
onExecute	onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onHeader onScriptInit onRecord	onFilterInit onFilterSave onFilterValidate	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onFooter onHeader onScriptInit onRecord

sc_send_notification('title', 'message', 'destiny_type', 'to', 'from', 'link', 'dtexpire', 'profile')

The macro sc_send_notification allows sending notifications to system users dynamically.

Parameter Description

Parameter	Description	Example
title =>	**Mandatory parameter** that receives the title of the notification that will be sent. The text can be provided in quotes, using variables or langs.	Using variable $title_not = "message title"; 'title' => $title_not Direct text in the macro or with lang 'title' => 'message title' or 'title' => {lang_notificat
message =>	**Mandatory parameter** that receives the body of the notification message that will be sent. This parameter allows the use of texts with or without HTML formatting and can be provided in quotes, variables, or lang.	Using variable $msg_not = "Notification message body"; 'message' => $msg_not Direct text in the macro 'message' => 'Notification message body'
destiny_type =>	**Mandatory parameter** that defines what data must be provided in the **to** parameter for sending the notification. The values that must be provided are: • **'user'** - Defines that the **to** parameter must receive the **login** of the users who should receive the notification message. • **'group'** - Defines that the **to** parameter must receive the names of the groups, from the security module, that should receive the notification message. **This option is only available in systems that use the security module by group.** • **'all'** - Defines that all system users should receive the notification message. **In this case, the to parameter must be passed as empty.** • **'profile'** - Defines that the **to** parameter must receive the names of the profiles created in the notification system, in the Send Notification application, that should receive the message.	Example of how the parameter should be p 'destiny_type' => 'user' Accepted values are: **user**, **group**, **all** and
to =>	Defines the system users who should receive the notification, within the universe defined in the destiny_type parameter. • **When specifying destiny_type =>'user'** - The developer must provide a list of the logins of the users who should receive the notification. The provided values must be separated by semicolons (;). • **When specifying destiny_type => 'group'** - The developer must provide the list of user groups, from the security module, that should receive the notification. • **When specifying destiny_type => 'all'** - In this case, all system users should receive the notification, and the **to** parameter must receive an empty value. • **destiny_type => 'profile'** - The developer must provide the names of the profiles created in the notification system, in the Send Notification application, that should receive the message. In all cases, the provided values must be separated by semicolons (;)	Example when using **destiny_type =>'us** **'to' => 'joao;pedro;carlos'** Example when using **destiny_type =>'gr** 'to' => 'employees;administrators' Example when using **destiny_type =>'all** 'to' => '' Example when using **destiny_type =>'pr** 'to' => 'sending_group_name1;sending_group_name
from =>	**Mandatory parameter** that defines the system user sending the notification. This parameter accepts only one value at a time and can be defined by directly providing a user's name or through a variable.	'from' => 'admin'
link =>	Optional parameter that receives a link to access an application in the system or an external URL. The text can be provided in quotes, using variables.	Providing an external URL 'link' => 'https://scriptcase.com.br/' Providing an application from the system 'link' => 'app_name'
dtexpire =>	Optional parameter that defines the date and time when the notification will expire and will no longer be displayed in the quick access icon of the menu. Upon reaching the date, the user will only be able to access the message through the inbox application. The date and time format must be: YYYY-MM-DD HH:MM:SS	'dtexpire' => '2024-10-02 23:59:59'
profile =>	**Mandatory parameter** that must receive the name of the profile that the developer specified when creating the notification module in the Scriptcase interface.	'profile' => 'samples_notification'

Example

```
$title = 'New user';
$msg = 'User: ' . $struser . ' registered in the system';

sc_send_notification([
    'title' => $title,
    'message' => $msg,
    'destiny_type' => 'user',
    'to' => 'admin',
    'from' => 'admin',
    'link' => '',
    'dtexpire' => '2024-10-02 23:59:59',
    'profile' => 'samples_profile_name',
]);
```

sc_send_notification(array $parameters = $_sc_settings)

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree me

Watch a tutorial

This macro sends SMS messages using APIs available in the tool.

- clickatell
- plivo
- twilio

Numbers must include the country code.

Parameters

Parameter	Description
$arr_settings	The macro has only one parameter, which must receive an array containing two arrays: one for configurations and another for the message.

Array Index	Description
from	Origin number of the message
to	SMS recipient
message:	The text message content
gateway:	API to be used (clickatell, plivo, or twilio)
auth_token:	API authentication token

Examples

Example 1 - Basic structure for sending using clickatell:

```
$arr_settings = array(
 'message' =>
 [
  'from' => '+55999999999', //origin number (Required)
  'to' => '+55999999999', //destination number (Required)
  'message' => 'Scriptcase SMS' //Message to be sent (Required)
 ],
 'settings' =>
 [
  'gateway' => 'clickatell', //sending gateway (Required)
  'auth_token' => '54Cik1-gQ4GqtuN70d4y4w==' //API token (Required)
 ]
);

sc_send_sms($arr_settings);
```

Example 2 - Basic structure for sending using Plivo:

```
$arr_settings = array(
 'message' =>
 [
    'from' => '+55999999999', //origin number (Required)
    'to' => '+5581984796338', //destination number (Required)
    'message' => 'Scriptcase SMS' //Message to be sent (Required)
 ],
 'settings' =>
 [
    'gateway' => 'plivo', //sending gateway (Required)
    'auth_id' => 'MAMDHIYZM0YJCZYTJLNJ', //Authentication token (Required)
    'auth_token' => 'ZGExOTgzZjNiOTE4YjBiN2ExMzdlZGYxYzQ4Yjc1' //API token (Required)
 ]
);

sc_send_sms($arr_settings);
```

Example 3 - Basic structure for sending using Twilio:

```
$arr_settings = array(
 'message' =>
 [
    'from' => '+55999999999', //origin number (Required)
    'to' => '+55999999999', //destination number (Required)
    'message' => 'Scriptcase SMS' //Message to be sent (Required)
    //'mediaURL' => 'http://royalsalutre.netmake.com.br/tools/nm/sc_helpv9/gerado/dev/pt_br/manual_mp/assets/images/scriptcase-logo.svg', //Allows sending an image
 ],
 'settings' =>
 [
    'gateway' => 'twilio', //sending gateway (Required)
    'auth_id' => 'MAMDHIYZM0YJCZYTJLNJ', //Authentication token (Required)
    'auth_token' => 'ZGExOTgzZjNiOTE4YjBiN2ExMzdlZGYxYzQ4Yjc1' //API token (Required)
 ]
);

sc_send_sms($arr_settings);
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Tree menu	ReportPDF application

sc_seq_register

Watch a tutorial

This macro provides the record (being processed) grid sequential number. It can be used, only in **onRecord** event.

Ex. 1:
```
if({sc_seq_register} == 10) {
// last line of the page
{total} = {sum_total};
}
```

Macro Scope

Grid application
onRecord

sc_set_export_name ("export_type", "file_name")

The files resulting from the exports made in the query are stored in thedirectory **tmp** (temporary)specified in the tool's settings.

Both in development and in production, files are stored with theprefix **sc_** and a unique key. After a while, files with the prefix **sc_** will be deleted.

To prevent the exported files from being deleted, it is possible to use this macro to rename them.

Parameter List

Parameter	Value	Description
"export_type"	String containing the export type, which must be protected by quotation marks. Example of use in the macro. **"csv"** **List of supported values:** • csv • json • pdf • rtf • xml • xls • word	**Mandatory parameter**, where we must inform the type of file we want to rename. **This parameter does not accept variables.**
"filename"	String containing the name the file will receive. Example of use in the macro: **"New file name"** **Supported values** • String with the name • Global variable	Mandatory parameter, where we must inform the name that the file will receive.

See below some examples of using the macro:

Example 1 - Defining a name for a PDF file

```
sc_set_export_name ("pdf", "meu_arquivo_pdf.pdf");
```

Example 2 - Defining a file name by passing the value through a global variable.

```
sc_set_export_name ("json", [glo_nome_json]);
```

Macro Scope

chart application	Grid application
onScriptInit	onApplicationInit onScriptInit

sc_set_fetchmode(parm);

Watch a tutorial

This macro allows to change the type of return from the dataset of the select commands.
The macro will processed before running the SQL command. Informing the parameter, the dataset will return the index or the name of the column.

parm = 0 : Returns an array with the index and the name of columns (Scriptcase's Default)
parm = 1 : Returns an array only with indexes of the columns

Example 1: Returning an array with the name of the columns.

```
sc_set_fetchmode(0);
sc_select(my_data, "SELECT customerid, stateid, birthdate, creditlimit FROM customers where customerid = 'ALFKI'");

while (!$my_data->EOF){
$meus_dados->fields['customerid'];
$customer = $my_data->fields['stateid'];
$customer = $my_data->fields['birthdate'];

$my_data->MoveNext();
}
$my_data->Close();
```

Example 2: Returning an array with the index of the columns.

```
sc_set_fetchmode(1);
sc_select(my_data, "SELECT customerid, stateid, birthdate, creditlimit FROM customers where customerid = 'ALFKI'");

while (!$my_data->EOF){
$my_data->fields[1];
$customer = $my_data->fields[2];
$customer = $my_data->fields[3];

$my_data->MoveNext();
}
$my_data->Close();
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu	Reportl applica
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onExecute onLoad	onExecute onLoad	onExecute onLoad	onFoote onHeade onScript onRecor

sc_set_focus('Field')

Watch a tutorial

This macro is used to set the focus to a specific field in a form application.

Ex. 1:

sc_set_focus('name');

Macro Scope

calendar application	Form application	Control Form
OnClick onLoad onRefresh onValidateSuccess	OnClick onLoad onRefresh onValidateSuccess	onApplicationInit onScriptInit onLoadAll onRefresh

sc_set_global($variable_01) or ({My_Field})

Watch a tutorial

This macro is used to register global variables. Create a session variable with the same name and content of the local variable.

NOTE: This macro will be discontinued soon. You should use the method of creating global variables using brackets. Ex: [var_glo_user] = "test";

Ex. 1: Registering a users variable.
$my_var = "mary";
sc_set_global($my_var);

Creates a session variable with the name "my_var" with the "mary" content.

ï¿½

Ex. 2: To register a field value as a session variable.
sc_set_global({fieldname});

Note: *This macro doesn't attribute values. Only register the session variables from PHP.*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Respons Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarApplicationInit onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplica onExecut onLoad

sc_set_groupby_rule

Watch a tutorial

This macro is used to select a GROUP BY rule in execution time in the onApplicationInit event.

Ex:

sc_set_groupby_rule = "rule id";

Macro Scope

Grid application
onApplicationInit

sc_set_language('String Language')

Watch a tutorial

This macro allows us to define the application language dynamically.

Ex. 1: Changes the language to English.
sc_set_language('en_us');

Ex. 2: **Changes the language to Spanish.**
sc_set_language('es');

It still allows to define the language and regional settings of the applications using a delimiter ";" between the parameters.

Ex. 1: Changing the language to English and regional settings to Canada.
sc_set_language('en_us;en_ca');

Ex. 2: Changing the language to Spanish and regional settings to Mexico.
sc_set_language('es;es_mx');

Ex. 3: Changing the language to Germanian and regional settings to Luxembourg.
sc_set_language('de;de_lu');

Ex. 4: Changing the language to Slovak and regional settings to Slovak Republic.
sc_set_language('sk;sk_sk');

Languague Code	
Language	**Value**
Arabic	ar
Bahasa Melayu	ms
Belarusian	be
Bengali	bn
Bosnian	bs
Bulgarian	bg
Catalan	ca
Chinese Simplified	zh_cn
Chinese Traditional	zh_hk
Croatian	hr
Czech	cz
Danish	da
Dutch	nl
English	en_us
Estonian	et
Finnish	fi
French	fr
Galician	gl
German	de
Greek	el
Gujarati	gu
Hebrew	he
Hindi	hi
Hungarian	hu_hu
Indonesian	id
Italian	it
Japanese	ja
Kannada	kn
Korean	ko
Latvian	lv
Lithuanian	lt
Macedonian	mk
Marathi	mr
Norwegian	no

Regional settings code	
Reginal Settings	**Value**
Afrikaans (South Africa)	af_za
Albanian (Albania)	sq_al
Amharic (Ethiopia)	am_et
Arabic (Algeria)	ar_dz
Arabic (Bahrain)	ar_bh
Arabic (Egypt)	ar_eg
Arabic (Iraq)	ar_iq
Arabic (Jordan)	ar_jo
Arabic (Kuwait)	ar_kw
Arabic (Lebanon)	ar_lb
Arabic (Libya)	ar_ly
Arabic (Morocco)	ar_ma
Arabic (Oman)	ar_om
Arabic (Qatar)	ar_qa
Arabic (Saudi Arabia)	ar_sa
Arabic (Syria)	ar_sy
Arabic (Tunisia)	ar_tn
Arabic (United Arab Emirates)	ar_ae
Arabic (Yemen)	ar_ye
Armenian (Armenia)	hy_am
Azerbaijan (Azerbaijan)	az_az
Bahasa Melayu (Malaysia)	ms_my
Basque	eu
Belarusian (Belarus)	be_by
Bengali (Bangladesh)	bn_bd
Bengali (India)	bn_in
Bosnian (Bosnia-Herzegovina)	bs_ba
Breton (France)	br_fr
Bulgarian (Bulgaria)	bg_bg
Catalan (Spain)	ca_es
Chinese (Republic of China)	zh_cn
Chinese (Simplified, Singapore)	zh_sg
Chinese (Traditional, Hong Kong)	zh_hk
Chinese (Traditional, Macau)	zh_mo

Polish	pl
Portuguese Brazil	pt_br
Portuguese Portugal	pt_pt
Punjabi	pa
Romanian	ro
Russian	ru
Serbian Cyrillic	sr
Shuar	jiv
Sinhalese	si
Slovak	sk
Slovenian	sl
Spanish	es
Swedish	sv
Telugu	te
Thai	thai
Turkish	tr
Ukrainian	uk
Urdu	ur

Chinese (Traditional, Taiwan)	zh_tw
Croatian (Bosnia and Herzegovina)	hr_ba
Croatian (Croatia)	hr_hr
Czech (Czech Republic)	cs_cz
Danish (Denmark)	da_dk
Dutch (Belgium)	nl_be
Dutch (Netherlands)	nl_nl
English (Australia)	en_au
English (Belize)	en_bz
English (Botswana)	en_bw
English (Cameroon)	en_cm
English (Canada)	en_ca
English (Caribbean)	en_cb
English (India)	en_in
English (Ireland)	en_ie
English (Jamaica)	en_jm
English (Malaysia)	en_my
English (Namibia)	en_na
English (New Zealand)	en_nz
English (Nigeria)	en_ng
English (Philippines)	en_ph
English (Singapore)	en_sg
English (South Africa)	en_za
English (Trinidad and Tobago)	en_tt
English (Uganda)	en_ug
English (United Kingdom)	en_gb
English (United States)	en_us
English (Zambia)	en_zm
English (Zimbabwe)	en_zw
Estonian (Estonia)	et_ee
Faroese (Faroe Islands)	fo_fo
Finnish (Finland)	fi_fi
French (Belgium)	fr_be
French (Cameroon)	fr_cm
French (Canada)	fr_ca
French (France)	fr_fr
French (Ivory Coast)	fr_ci
French (Luxembourg)	fr_lu
French (Monaco)	fr_mc
French (Switzerland)	fr_ch
Frisian, Western (Netherlands)	fy_nl
Galician (Spain)	gl_es
Georgian (Georgia)	ka_ge
German (Austria)	de_at
German (Germany)	de_de
German (Lichtenstein)	de_li
German (Luxembourg)	de_lu
German (Switzerland)	de_ch
Greek (Greece)	el_gr
Gujarati (India)	gu_in
Hausa (Ghana)	ha_gh
Hebrew (Israel)	he_il
Hindi (India)	hi_in
Hungarian (Hungary)	hu_hu
Icelandic (Iceland)	is_is
Indonesian (Indonesia)	id_id
Irish (Ireland)	ga_ie
Italian (Italy)	it_it
Italian (Switzerland)	it_ch
Japanese (Japan)	ja_jp
Kalaallisut (Kalaallit Nunaat)	kl_gl
Kannada (India)	kn_in
Kazakh (Kazakhstan)	kk_kz
Khmer (Cambodia)	km_kh
Kirghiz (Kyrgyzstan)	ky_kg
Korean (Korea)	ko_kr
Lao (Laos)	lo_la
Latvian (Latvia)	lv_lv
Lithuanian (Lithuania)	lt_lt
Luxembourgish (Luxembourg)	lb_lu
Macedonian (Republic of Macedonia)	mk_mk
Malayalam (India)	ml_in
Marathi (India)	mr_in
Mongolian (Mongolia)	mn_mn
Nepali (Nepal)	ne_np
Northern Sotho (South Africa)	nso_z
Norwegian (Norway)	no_no
Norwegian Bokmal (Norway)	nb_no
Norwegian Nynorsk (Norway)	nn_no
Occitan (France)	oc_fr
Oriya (India)	or_in

Persian (Iran)	fa_ir
Polish (Poland)	pl_pl
Portuguese (Angola)	pt_ao
Portuguese (Brazil)	pt_br
Portuguese (Portugal)	pt_pt
Punjabi (India)	pa_in
Romanian (Moldova)	ro_md
Romanian (Romania)	ro_ro
Russian (Belarus)	ru_by
Russian (Kazakhstan)	ru_kz
Russian (Russia)	ru_ru
Serbian Cyrillic (Serbia and Montenegro)	sr_yu
Serbian Latin (Serbia and Montenegro)	sh_yu
Slovak (Slovak Republic)	sk_sk
Slovenian (Slovenia)	sl_si
Sotho (South Africa)	st_za
Spanish (Argentina)	es_ar
Spanish (Bolivia)	es_bo
Spanish (Chile)	es_cl
Spanish (Colombia)	es_co
Spanish (Costa Rica)	es_cr
Spanish (Dominican Republic)	es_do
Spanish (Ecuador)	es_ec
Spanish (El Salvador)	es_sv
Spanish (Guatemala)	es_gt
Spanish (Honduras)	es_hn
Spanish (Mexico)	es_mx
Spanish (Nicaragua)	es_ni
Spanish (Panama)	es_pa
Spanish (Paraguay)	es_py
Spanish (Peru)	es_pe
Spanish (Puerto Rico)	es_pr
Spanish (Spain)	es_es
Spanish (United States)	es_us
Spanish (Uruguay)	es_uy
Spanish (Venezuela)	es_ve
Sri Lanka (Sinhalese)	si_si
Swahili (Tanzania)	sw_tz
Swedish (Finland)	sv_fi
Swedish (Sweden)	sv_se
Tajik (Tajikistan)	tg_tj
Tamil (India)	ta_in
Telugu (India)	te_in
Thai (Thailand)	th_th
Tswana (South Africa)	tn_za
Turkish (Turkey)	tr_tr
Ukrainian (Ukraine)	uk_ua
Urdu (Pakistan)	ur_pk
Uzbek (Uzbekistan)	uz_uz
Vietnamese (Vietnam)	vi_vn
Welsh (United Kingdom)	cy_gb
Xhosa (South Africa)	xh_za
Zulu (South Africa)	zu_za

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tr
onExecute	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit onFilterValidate	onApplicationInit onScriptInit	onApplicationInit onScriptInit onValidate onValidateSuccess	onApplicationInit onLoad	onApplicationInit onLoad	on on

sc_set_regional('String Regional')

Watch a tutorial

This macro allows to dynamically change the application regional settings.

Ex. 1:
sc_set_regional('en_us');

Language	Value
Afrikaans (South Africa)	af_za
Albanian (Albania)	sq_al
Amharic (Ethiopia)	am_et
Arabic (Algeria)	ar_dz
Arabic (Bahrain)	ar_bh
Arabic (Egypt)	ar_eg

Arabic (Iraq)	ar_iq
Arabic (Jordan)	ar_jo
Arabic (Kuwait)	ar_kw
Arabic (Lebanon)	ar_lb
Arabic (Libya)	ar_ly
Arabic (Morocco)	ar_ma
Arabic (Oman)	ar_om
Arabic (Qatar)	ar_qa
Arabic (Saudi Arabia)	ar_sa
Arabic (Syria)	ar_sy
Arabic (Tunisia)	ar_tn
Arabic (United Arab Emirates)	ar_ae
Arabic (Yemen)	ar_ye
Armenian (Armenia)	hy_am
Azerbaijan (Azerbaijan)	az_az
Bahasa Melayu (Malaysia)	ms_my
Basque	eu
Belarusian (Belarus)	be_by
Bengali (Bangladesh)	bn_bd
Bengali (India)	bn_in
Bosnian (Bosnia-Herzegovina)	bs_ba
Breton (France)	br_fr
Bulgarian (Bulgaria)	bg_bg
Catalan (Spain)	ca_es
Chinese (Republic of China)	zh_cn
Chinese (Simplified, Singapore)	zh_sg
Chinese (Traditional, Hong Kong)	zh_hk
Chinese (Traditional, Macau)	zh_mo
Chinese (Traditional, Taiwan)	zh_tw
Croatian (Bosnia and Herzegovina)	hr_ba
Croatian (Croatia)	hr_hr
Czech (Czech Republic)	cs_cz
Danish (Denmark)	da_dk
Dutch (Belgium)	nl_be
Dutch (Netherlands)	nl_nl
English (Australia)	en_au
English (Belize)	en_bz
English (Botswana)	en_bw
English (Cameroon)	en_cm
English (Canada)	en_ca
English (Caribbean)	en_cb
English (India)	en_in
English (Ireland)	en_ie
English (Jamaica)	en_jm
English (Malaysia)	en_my
English (Namibia)	en_na
English (New Zealand)	en_nz
English (Nigeria)	en_ng
English (Philippines)	en_ph
English (Singapore)	en_sg
English (South Africa)	en_za
English (Trinidad and Tobago)	en_tt
English (Uganda)	en_ug
English (United Kingdom)	en_gb
English (United States)	en_us
English (Zambia)	en_zm
English (Zimbabwe)	en_zw
Estonian (Estonia)	et_ee
Faroese (Faroe Islands)	fo_fo
Finnish (Finland)	fi_fi
French (Belgium)	fr_be
French (Cameroon)	fr_cm
French (Canada)	fr_ca
French (France)	fr_fr
French (Ivory Coast)	fr_ci
French (Luxembourg)	fr_lu
French (Monaco)	fr_mc
French (Switzerland)	fr_ch
Frisian, Western (Netherlands)	fy_nl
Galician (Spain)	gl_es
Georgian (Georgia)	ka_ge
German (Austria)	de_at
German (Germany)	de_de
German (Lichtenstein)	de_li
German (Luxembourg)	de_lu

German (Switzerland)	de_ch
Greek (Greece)	el_gr
Gujarati (India)	gu_in
Hausa (Ghana)	ha_gh
Hebrew (Israel)	he_il
Hindi (India)	hi_in
Hungarian (Hungary)	hu_hu
Icelandic (Iceland)	is_is
Indonesian (Indonesia)	id_id
Irish (Ireland)	ga_ie
Italian (Italy)	it_it
Italian (Switzerland)	it_ch
Japanese (Japan)	ja_jp
Kalaallisut (Kalaallit Nunaat)	kl_gl
Kannada (India)	kn_in
Kazakh (Kazakhstan)	kk_kz
Khmer (Cambodia)	km_kh
Kirghiz (Kyrgyzstan)	ky_kg
Korean (Korea)	ko_kr
Lao (Laos)	lo_la
Latvian (Latvia)	lv_lv
Lithuanian (Lithuania)	lt_lt
Luxembourgish (Luxembourg)	lb_lu
Macedonian (Republic of Macedonia)	mk_mk
Malayalam (India)	ml_in
Marathi (India)	mr_in
Mongolian (Mongolia)	mn_mn
Nepali (Nepal)	ne_np
Northern Sotho (South Africa)	nso_z
Norwegian (Norway)	no_no
Norwegian Bokmal (Norway)	nb_no
Norwegian Nynorsk (Norway)	nn_no
Occitan (France)	oc_fr
Oriya (India)	or_in
Persian (Iran)	fa_ir
Polish (Poland)	pl_pl
Portuguese (Angola)	pt_ao
Portuguese (Brazil)	pt_br
Portuguese (Portugal)	pt_pt
Punjabi (India)	pa_in
Romanian (Moldova)	ro_md
Romanian (Romania)	ro_ro
Russian (Belarus)	ru_by
Russian (Kazakhstan)	ru_kz
Russian (Russia)	ru_ru
Serbian Cyrillic (Serbia and Montenegro)	sr_yu
Serbian Latin (Serbia and Montenegro)	sh_yu
Slovak (Slovak Republic)	sk_sk
Slovenian (Slovenia)	sl_si
Sotho (South Africa)	st_za
Spanish (Argentina)	es_ar
Spanish (Bolivia)	es_bo
Spanish (Chile)	es_cl
Spanish (Colombia)	es_co
Spanish (Costa Rica)	es_cr
Spanish (Dominican Republic)	es_do
Spanish (Ecuador)	es_ec
Spanish (El Salvador)	es_sv
Spanish (Guatemala)	es_gt
Spanish (Honduras)	es_hn
Spanish (Mexico)	es_mx
Spanish (Nicaragua)	es_ni
Spanish (Panama)	es_pa
Spanish (Paraguay)	es_py
Spanish (Peru)	es_pe
Spanish (Puerto Rico)	es_pr
Spanish (Spain)	es_es
Spanish (United States)	es_us
Spanish (Uruguay)	es_uy
Spanish (Venezuela)	es_ve
Sri Lanka (Sinhalese)	si_si
Swahili (Tanzania)	sw_tz
Swedish (Finland)	sv_fi
Swedish (Sweden)	sv_se
Tajik (Tajikistan)	tg_tj

Tamil (India)	ta_in
Telugu (India)	te_in
Thai (Thailand)	th_th
Tswana (South Africa)	tn_za
Turkish (Turkey)	tr_tr
Ukrainian (Ukraine)	uk_ua
Urdu (Pakistan)	ur_pk
Uzbek (Uzbekistan)	uz_uz
Vietnamese (Vietnam)	vi_vn
Welsh (United Kingdom)	cy_gb
Xhosa (South Africa)	xh_za
Zulu (South Africa)	zu_za

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tr
onExecute	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit onFilterValidate	onApplicationInit onScriptInit	onApplicationInit onScriptInit onValidate onValidateSuccess	onApplicationInit onLoad	onApplicationInit onLoad	on on

sc_set_theme('String Theme')

Watch a tutorial

This macro allows to dynamically change the theme that will be used in the applications.

When used, the theme is changed in the session, causing all applications accessed later to receive the informed theme.

Parameter

This macro has only one parameter, which must be named after the theme that will be applied.
This parameter can be passed in two ways.

Example 1 - String with the name of the theme.

```
sc_set_theme ('BlueBerry');
```

Example 2 - Local variable receiving the value of a field or a global variable.

```
$ set_theme = {field_name};
sc_set_theme ($ set_theme);
```

Related Links
See how to dynamically change the theme of a project using the macro.

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tr
onExecute	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit onFilterValidate	onApplicationInit onScriptInit	onApplicationInit onScriptInit onValidate onValidateSuccess	onApplicationInit onLoad	onApplicationInit onLoad	on on

sc_site_ssl

Watch a tutorial

This macro is used to verify if the used site is a safe site. (https protocol)

Ex. 1: Using an affirmative.
```
if (sc_site_ssl)
{
echo "ok - Safe site";
}
```

Ex. 2: Using a negative.
```
if (!sc_site_ssl)
{
echo "Warning - Unsafe site";
}
```

Ex. 3: Redirecting if the site isn't safe.
```
if (!sc_site_ssl)
{
sc_redir("http://www.erro_page.com/") ;
}
```

Note: *This macro works only on the web server IIS.*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tre me
onExecute	onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onScriptInit	onScriptInit	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onClick onScriptInit onLoadAll onValidate onValidateFailure onValidateSuccess	onApplicationInit onLoad	onApplicationInit onLoad	onL

sc_sql_injection({My_Field}) or ($My_Variable)

Watch a tutorial

This macro is used to protect the field/variable against "SQL injection" attempts.

Macro used for protection against "SQL injection" in commands generated by the developer when using the macros: sc_lookup, sc_select, or sc_exec_sql.

Ex. 1: Protecting a local variable:
$field_protect = sc_sql_injection({my_field});

Ex. 2: Protecting an user variable:
$field_protect = sc_sql_injection($my_var);

Note: *that all database accesses, generated for the Scriptcase, have protection against "sql injection".*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu	ReportI applica
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onScriptInit onWidgetLoad	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onExecute onLoad	onExecute onLoad	onExecute onLoad	onFoote onHeade onScript onRecor

sc_sql_protect(Value, "Type", "Connection")

Watch a tutorial

This macro will protect the value passed as a parameter according to the user database, the "Connection" parameter is optional, if not informed Scriptcase will use the default connection.

The "**Type**" parameter so far can only use the "**Date**" value.

Ex 01: In this example we will protect the dates passed as parameter so this way it can be correctly interpreted to the ACCESSED database.

```
sc_select_where(add) = " AND news_noticias.noticia_data_pub
BETWEEN ".sc_sql_protect($data_inicial, "date")." AND ".sc_sql_protect($data_final, "date")."";
```

Macro Scope

Blank application	calendar application	Grid application	Search application	Form application	Control Form	ReportPDF application
onExecute	onValidate	onClick onFooter onGroupBy onScriptInit onNavigate onRecord	onFilterValidate	onAfterDeleteAll onAfterInsertAll onAfterUpdateAll onBeforeDeleteAll onBeforeInsertAll onBeforeUpdateAll onValidate	onLoadAll onValidate	onRecord

sc_statistic (arr_val, tp_var)

This macro calculates and returns an array with the reported statistical values from an array of numeric values.

Two parameters are required for using the macro.

Parameter	Value	Description
arr_val	This parameter accepts only variables. **Example of use:** $arr_val = array (5230.01,8374,625.45,8926.34,7000,523.78); sc_statistic ($ arr_val, **$ type**);	Variable containing an array with the numeric input values. **Parameter required.**
tp_var	This parameter accepts the value or a variable. **Example using value:** sc_statistic ($arr_val, '**P**'); **Example using a variable:** $type = 'P'; sc_statistic ($arr_val, **$type**); List of values accepted in the parameter. • **P** • **A**	This parameter defines the type of calculation of the **variance**(index 2) and **standard deviation**(index 3) of the array. Accepted values are **P** and **A** • **P** = Population • **A** = Sampling **Parameter required.**

This macro returns the calculation in an array with 10 positions, where each position is related to a statistical data.

Check below what each index will return.

Values returned in each index of the array:

Index	Value
[0]	Arithmetic mean
[1]	Median
[2]	Variance
[3]	Standard deviation
[4]	Amplitude
[5]	Number of different values
[6]	Number of array values
[7]	Number of Null values
[8]	Lowest value
[9]	Highest value

Example of using the macro

```
$arr_val = array (5230.01.8374,625.45.8926.34,7000,523.78,523.78,6897.21,987.56,987.56,8729,63);

$arr_stat = sc_statistic ($arr_val, 'P');
```

Macro return:

```
Array
(
    [0] => 4072.3075
    [1] => 3108.785
    [2] => 12817792.489302
    [3] => 3580.1944764638
    [4] => 8863.34
    [5] => 10
    [6] => 12
    [7] => 12
    [8] => 63
    [9] => 8926.34
);
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Tree menu	ReportPDF application

					ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess				

sc_text_style({my_field}, "Background-Color", "Size", "Color", "Family", "Weight")

Watch a tutorial

This macro allows us to modify the text style of a grid field dynamically.

Parameter	Description
Field	Name
Background-Color (optional)	Text cell background Color
Size (optional)	Text font size
Color (optional)	Text Color
Family (optional)	Text family
Weight (optional)	Text weight

Ex. 1:
sc_text_style({my_field}, '#33FF99', '15px', '#000000', 'Arial, sans-serif', 'bold');

Macro Scope

Grid application
onRecord

sc_time_diff({datetime_01}, "Date_01 Format", {datetime_02}, "Date_02 Format")

Watch a tutorial

sc_time_diff({datetime_01}, "Date_01 Format", {datetime_02}, "Date_02 Format");

This macro calculates the difference between hour, minutes, and seconds for two DateTime/time values.

The result will be returned in an array, with the dimensions [0], [1] and [2] containing hours, minutes, and seconds, respectively.

Parameter	Description
datetime_01	Value or variable for datetime_01
Date_01 Format	Value or variable to specify the format of datetime_01
datetime_02	Value or variable for datetime_02
Date_02 Format	Value or variable to specify the format of datetime_02

Ex. 1: General Format

```
{differences} = sc_time_diff({datetime_01}, "Date_01 Format", {datetime_02}, "Date_02 Format");

{diff_hours} = {differences[0]};
{diff_minutes} = {differences[1]};
{diff_seconds} = {differences[2]};
```

Ex. 2: Returning positive values

```
{differences} = sc_time_diff("2012-07-25 05:33:45", "yyyy-mm-dd hh:ii:ss", "2012-07-21 15:22:57", "yyyy-mm-dd hh:ii:ss");

{differences[0]} would be equal to 86 (hours);
{differences[1]} would be equal to 10 (minutes);
{differences[2]} would be equal to 48 (seconds);
```

Ex. 3: Returning negative values

```
{differences} = sc_time_diff("2012-07-21 15:22:57", "yyyy-mm-dd hh:ii:ss", "2012-07-25 05:33:45", "yyyy-mm-dd hh:ii:ss");

{differences[0]} would be equal to -86 (hours);
{differences[1]} would be equal to -10 (minutes);
{differences[2]} would be equal to -48 (seconds);
```

Ex. 4: Considering only the hours' format - Positive return

```
{differences} = sc_time_diff("18:14:55", "hh:ii:ss", "10:55:22", "hh:ii:ss");

{differences[0]} would be equal to 7 (hours);
{differences[1]} would be equal to 19 (minutes);
{differences[2]} would be equal to 33 (seconds);
```

Ex. 5: Considering only the hours' format - Negative return

```
{differences} = sc_time_diff ("10:55:22", "hh:ii:ss", "18:14:55", "hh:ii:ss");

{differences[0]} would be equal to -7 (hours);
{differences[1]} would be equal to -19 (minutes);
{differences[2]} would be equal to -33 (seconds);
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_trunc_num({My_Field}, Decimal_Number)

Watch a tutorial

This macro is used to truncate numerical values that exceed the decimals amount specified.

Parameter	Descri?003f006f
My_Field	Variable with value to format (return in the same variable).
Decimal_Number	Amount of decimals to display.

Ex. 1:
sc_trunc_num({my_value}, 2);

Input value = 1250.235
Output value = 1250.23

Input value = 1250.2
Output value = 1250.20

Input value = 1250
Output value = 1250.00

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onClick onApplicationInit onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad

sc_url_exit(URL)

Watch a tutorial

This macro modifies the application exit URL.

Ex. 1:
sc_url_exit(http://www.scriptcase.net);

Ex. 2:
sc_url_exit(aplx.php);

Macro Scope

calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	ReportPDF application
onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onScriptInit	onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onClick onScriptInit onLoadAll onValidate onValidateFailure onValidateSuccess	onRecord

sc_url_library("Target", "Library Name", "File")

Watch a tutorial

This macro will return the file path, inside a library, to be used on the Scriptcase applications. It is necessary to have a library created on Scriptcase to use this Macro. To create a library, go to the "Tools > Library" Menu.

Parameter	Description
Target	Tells you what the scope of the library. It can be "sys" for libraries of *Public* scope or "prj" for libraries of the *Project* scope.
Library Name	Given name to the library at the time of creation.
File	The absolute path within the library.

Ex. 1 - Importing CSS files from a library:
<*link rel="stylesheet" type="text/css" href="<*?php echo sc_url_library('prj', 'bootstrap-3.3.2-dist', 'css/style1.css'); ?>" />

Ex. 2 - Importing a JS file from a library:
<*script type="text/javascript" src="<*?php echo sc_url_library('prj', 'bootstrap-3.3.2-dist', 'js/bootstrap,js'); ?>">

NOTE: *Note that you need to remove the asterisk (*) form the examples above to use it.*

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onCalendarApplicationInit onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onScriptInit	onApplicationInit onScriptInit onWidgetLoad	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplication onExecute onLoad

sc_user_logout('variable_name', 'variable_content', 'apl_redir.php', 'target')

This macro is used to log the user out to the system.

If the target is not informed, the default value is "_top".

Ex:

sc_user_logout('usr_login', 'admin', 'ctrl_login.php', '_self');

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Form application	Control Form	ReportPDF application
onExecute	onAfterDelete onAfterInsert onAfterUpdate onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onValidate onValidateSuccess	onHeader onScriptInit	onScriptInit	onScriptInit	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onScriptInit onLoad onNavigate onLoadRecord onValidate onValidateSuccess	onScriptInit onLoadAll onValidate onValidateSuccess	onHeader onScriptInit onRecord

sc_warning'on' or 'off'

Watch a tutorial

This macro dynamically enables or disables the display of warning messages.

Warning messages are generated in situations such as referencing a non-existent variable, an undefined array item, or other similar conditions. This does not resolve the underlying issue but hides or shows the warnings.

Ex. 1: To deactivate the message (do not display).
sc_warning = 'off';

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tre
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onScriptInit	onApplicationInit	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onA onL onL

sc_webservice("Method", "URL", "Port", "Send Method", "Parameters Array", "Setting's Array", "Timeout", "Return")

Watch a tutorial

This macro is used to communicate with a web service.

Parameter	Description
Method	Informs what method will be used to access the web service. Ex: *file_get_contents* or *curl*.
URL	URL used to access the web service.
Port	Port used to access the web service.
Send Method	Defines the request method of the web service. It can be *GET* ou *POST*.
Parameters Array	Parameter's array used to access the web service, varying according to the accessed service.
Settings's Array(optional)	Parameter's array for configuration of the access method used.
Timeout(optional)	Service's access timeout.
Return(optional)	Web service's return method. When it is set as "True", the return from the web service comes as an array, if it is set as "False" it will return as the original web service format. Default: True

```
$parms = array(
'METHOD' => 'GetBalance',
'RETURNALLCURRENCIES' => 1,
'VERSION' => urlencode(51.0),
'USER' => urlencode({paypal_api_login}),
'PWD' => urlencode({paypal_api_password}),
'SIGNATURE' => urlencode({paypal_api_key})
);
$parms = http_build_query($parms);
```

Ex1 - Using the macro *file_get_contents*:

```
sc_webservice('file_get_contents', $webservice_url, $_SERVER["SERVER_PORT"], "POST",
$parms, array(), 30);
```

Ex2 - Using the macro with CURL:

```
$response = sc_webservice("curl", {paypal_type}, 80, "POST", $parms,
array(CURLOPT_RETURNTRANSFER => true, CURLOPT_SSL_VERIFYPEER=>false), 30);
```

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tre me
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onScriptInit	onApplicationInit onScriptInit	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onBlur onChange onClick onFocus onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onE onL

Watch a tutorial

Variable to reference the application where clause plus the filter selections (if any). Reflects the where clause currently used.

Ex. 1:
$save_current_where = {sc_where_current};

Macro Scope

chart application	Grid application	ReportPDF application
onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onFooter onHeader onScriptInit onRecord

sc_where_filter

Watch a tutorial

This macro is used to retrieve the content generated by the "filter", according to its selections. These are the data that will be added to the application original where clause after filter submission.

Ex. 1:
$save_current_filter = {sc_where_filter};

Macro Scope

chart application	Grid application	ReportPDF application
onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onFooter onHeader onScriptInit onRecord

sc_where_orig

Watch a tutorial

It allows the developer to recover the application's original WHERE clause, inserted directly into the query in the SQL menu or when creating the query.

Because it is a value recovery macro, it has a different syntax from other macros. we develop applications between curly brackets: {sc_where_orig}

> The value retrieved by the macro is not affected by filters performed in the application or any changes made to the application through the macro sc_select_where(add).

See below some examples of the macro

Example 1 - Retrieving the WHERE clause value from the application's SQL

Macro Scope

chart application	Grid application	Search application	ReportPDF application
onFooter onHeader onScriptInit	onClick onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onApplicationInit onFilterInit onFilterRefresh onFilterSave onFilterValidate	onFooter onHeader onScriptInit onRecord

sc_widget_config($arrayOptions)

The sc_widget_config() macro allows you to dynamically change the visual properties of the widget being executed. It must be used inside the onWidgetLoad event and applies settings only to the current widget.

This macro receives a single **associative array** containing the configuration options for the widget.

Example array structure:

```
sc_widget_config([
  'title' => 'My title',
  'border-color' => '#FF0000'
]);
```

Parameters by widget type

Available options vary depending on the type of widget being modified. See below for each type.

Link Widget

Parameter	Value	Description
title	'title' => 'Widget title'	Defines the title displayed at the top of the widget.
border-color	'border-color' => '#FF0000'	Border color of the widget, in hexadecimal format.

Divider Widget

Parameter	Value	Description
title	'title' => 'Main Section'	Title of the section created by the divider widget.
subtitle	'subtitle' => 'Complementary description'	Text displayed below the divider title.
background-color	'background-color' => '#F1F1F1'	Background color of the widget, in hexadecimal format.
border-color	'border-color' => '#CCCCCC'	Border color of the widget, in hexadecimal format.

Index Widget

Parameter	Value	Description
title	'title' => 'Sales Indicator'	Title displayed at the top of the index widget.
legend	'legend' => 'Compared to last month'	Text shown below the index's main value.
background-color	'background-color' => '#E0FFE0'	Background color of the widget, in hexadecimal format.
border-color	'border-color' => '#00CC00'	Border color of the widget, in hexadecimal format.

Full example

```
$widgetName = sc_widget_name();

if ($widgetName == 'sales_index') {
  sc_widget_config([
    'title' => 'Monthly Sales',
    'legend' => 'Compared to last month',
    'background-color' => '#f0f0f0',
    'border-color' => '#999999',
  ]);
}
```

In the example above, the sc_widget_config() macro is used to apply visual settings to the widget identified as sales_index.

Macro Scope

Dashboard
onWidgetLoad

sc_widget_data('dataName')

The sc_widget_data() macro returns data from an Index widget, allowing access to comparative and statistical values defined in the widget's configuration.

It can be used to customize the widget's content and appearance based on runtime-calculated data, such as percentage variation or difference between periods.

Parameter

This macro receives a single parameter, which must be the **name of the data** you want to retrieve.

Example of use:

```
sc_widget_data('value')
```

Available parameters

Parameter	Description
value	Main value of the current period.
period	Current period configured in the widget (e.g., current month, week, etc.).
last_value	Value of the previous period.
last_period	Label of the previous period (e.g., previous month, last week, etc.).
difference	Difference between the two periods (value - last_value).
variation	Percentage variation between periods (comparison between value and last_value).

Example

```
$dif = sc_widget_data('difference');

if ($dif < 0) {
  sc_widget_config([
    'background-color' => '#ffdddd',
    'border-color' => '#cc0000',
    'legend' => 'Performance below the previous period',
  ]);
} else {
  sc_widget_config([
    'background-color' => '#ddffdd',
    'border-color' => '#00cc00',
    'legend' => 'Improved compared to the previous period',
  ]);
}
```

In the example above, the sc_widget_data() macro retrieves the difference between periods. Based on that value, the widget's appearance is dynamically adjusted.

Macro Scope

Dashboard
onWidgetLoad

sc_widget_name

The sc_widget_name() macro returns the name of the widget currently being executed, allowing you to identify which widget is being processed and apply specific dynamic configurations.

This macro has no parameters.

Example

```
// Get the widget name to change its title
$widgetName = sc_widget_name();

// Check the widget name
if ($widgetName == 'widget1') {
    sc_widget_config([
        'title' => 'Sales Widget',
    ]);
}
```

In the example above, the sc_widget_name() macro is used to identify the widget being executed. If the name is widget1, the title is dynamically updated using the sc_widget_config() macro.

See also

For additional widget customization, see the macros: sc_widget_config() and sc_widget_data().

Macro Scope

Dashboard
onWidgetLoad

sc_zip_file("File", "Zip")

Watch a tutorial

This macro is used to generate ZIP files from a list of files and/or directories. The file parameter must be one of the following:

- Filename
- Directory name
- A variable containing an array, which contains files and/or directories list.
- A variable a file or a directory name.
- A variable containing an array, which contains files and/or directories list.

The zip parameter must contain the name of the zip file generated or the path to the file.

Ex. 1: Single file compressing.
sc_zip_file("/test/sample.htm", "/tmp/test.zip");

Ex. 2: Single directory compressing.
sc_zip_file("/test", "/tmp/test.zip");

Ex. 3: Multiple files and directories in an array.
$prep = array();
$prep[] = "/test/sample.htm";
$prep[] = "/test";
sc_zip_file($prep, "/tmp/test.zip");

Macro Scope

Blank application	calendar application	chart application	Grid application	Dashboard	Search application	Form application	Control Form	Menu application	Responsive Menu	Tre
onExecute	onAfterDelete onAfterInsert onAfterUpdate onApplicationInit onBeforeDelete onBeforeInsert onBeforeUpdate onScriptInit onLoad onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onFooter onHeader onScriptInit	onScriptInit	onApplicationInit	onApplicationInit onFilterInit onFilterSave onFilterValidate	onAfterDelete onAfterDeleteAll onAfterInsert onAfterInsertAll onAfterUpdate onAfterUpdateAll onApplicationInit onBeforeDelete onBeforeDeleteAll onBeforeInsert onBeforeInsertAll onBeforeUpdate onBeforeUpdateAll onClick onScriptInit onLoad onNavigate onLoadRecord onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onClick onScriptInit onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onLoad	onA onB onL

Authentication Variables

Watch a tutorial

These variables must be defined on applications with PDF export on environments with integrated authentication.

Variables	Description
[sc_pdf_usr]	Authentication User.
[sc_pdf_pw]	Authentication Password

Ex. 1: On a login application, it is possible to set these variables from the info given by the user:

[sc_pdf_usr] = {login_usr};
[sc_pdf_pw] = {senha_usr};

Macro Scope

Grid application	Form application
onApplicationInit onFooter onGroupBy onScriptInit	onScriptInit

Database Variables

Global variables containing database access values used by main connection.

Variables	Description
sc_glo_server	Server name.
sc_glo_db	Database name.
sc_glo_db_type	Database type (mssql, oracle, mysql, etc).
sc_glo_user	Connected user name.
sc_glo_pass	Database password access.
sc_glo_cript_pass	Encrypted password (S/N).
sc_glo_decimal_db	Used decimal separator (point or comma).

Ex. 1:
$my_database = [sc_glo_db];

These variables only reference the values without updating it (it is not possible to attribute/modify the values). To define user and the password to connect to the database, use the following variables:

Variables	Description
sc_db_master_usr	User login to be used.
sc_db_master_pass	Password to be used.
sc_db_master_cript	Encrypted password (S/N) (see sc_encode macro).

Important: *These variables only take effect during next application execution remaining valid during all the session (unless modified).*

Ex. 2: An application type "control" with a form created with information "login" and encrypted "password". These data must be used to connect to the database, during the system applications execution.
$temp_pass = sc_encode({password});
[sc_db_master_usr] = {login};
[sc_db_master_pass] = $temp_pass;
[sc_db_master_cript] = "S";

Macro Scope

Blank application	calendar application	chart application	Grid application	Search application	Form application	Control Form	Menu application	Responsive Menu	Tree menu	ReportPDF application
onExecute	ajaxFieldonBlur Onchange OnClick OnFocus	onFooter onHeader onScriptInit	onFooter onGroupBy onHeader onScriptInit onNavigate onRecord	onFilterInit onFilterRefresh onFilterSave onFilterValidate	ajaxFieldonBlur Onchange OnClick OnFocus	onBlur onChange onClick onLoadAll onRefresh onValidate onValidateFailure onValidateSuccess	onApplicationInit onExecute onLoad	onApplicationInit onExecute onLoad	onExecute onLoad	onFooter onHeader onScriptInit onRecord

Totalling Variables

During group by and summary processing, Scriptcase generates total variables to each group level.

The routines are defined in event “**onGroupBy**” determining the break levels and avalilable to each record. Totals special variables are then referenced substituting the group name for the key word “groupby”, as in:

Variables	Description
{count_ger}	Records total amount.
{sum_parcel}	Will show the total sum for the field "Parcel"
{sum_balance}	Will show the total sum for the field "Balance"
{count_groupby}	Current Group records total.
{sum_groupby_parcel}	Current Group level sum for the field “Parcel”.
{sum_groupby_balance}	Current Group level sum for the field “Balance”.

Ex. 1: In an application with a group by state and city that totalizes a balance field, in the group totals, is possible to display the average (avg) instead of the balance as follows.

{sum_groupby_balance} = {sum_groupby_balance} / {count_groupby};

Macro Scope

chart application	Grid application	ReportPDF application
onFooter onHeader	onFooter onGroupBy onRecord	onFooter onHeader onRecord

Totalling Variables (group by)

During the group by processing, Scriptcase generates total variables to each group level. The routines are defined in event “**onGroupBy**” determining the group levels. Totals special variables are then referenced substituting the group name for the key word “**quebra**”, as in:

Variable	Description
{count_ger}	Records total amount.
{sum_parcel}	Field “parcels” total.
{sum_balance}	Field “Balance” total.
{count_quebra}	Current group level records total.
{sum_quebra_parcel}	Current group level field “parcels” total.
{sum_quebra_balance}	Current group level field “Balance” total.

Ex. 1: In an application with a group by state and city and that totalizes a balance field, in the group totals, is possible to display the average (avg) instead of the balance as follows.

```
{sum_quebra_balance} = {sum_quebra_balance} / {count_quebra};
```

Macro Scope

Grid application
onFooter onGroupBy

Macro x Applications x Events

ajaxFieldonBlur

Aplicação Calendário	Aplicação Formulário	Aplicação Controle
sc_ajax_javascript	sc_ajax_javascript	sc_ajax_javascript sc_ajax_message
sc_ajax_message	sc_ajax_message	sc_begin_trans
sc_begin_trans	sc_begin_trans	sc_calc_dv
sc_calc_dv	sc_calc_dv	sc_commit_trans
sc_commit_trans	sc_commit_trans	sc_concat
sc_connection_edit	sc_connection_edit	sc_connection_edit
sc_connection_new	sc_connection_new	sc_connection_new
sc_date	sc_date	sc_date
sc_date_conv	sc_date_conv	sc_date_conv
sc_date_dif	sc_date_dif	sc_date_dif
sc_date_dif_2	sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_date_empty	sc_date_empty
sc_decode	sc_decode	sc_decode
sc_encode	sc_encode	sc_encode
sc_error_exit	sc_error_exit	sc_error_exit
sc_error_message	sc_error_message	sc_error_message
sc_exec_sql	sc_exec_sql	sc_exec_sql
sc_field_display	sc_field_display	sc_field_display
sc_field_readonly	sc_field_readonly	sc_field_readonly
sc_format_num	sc_format_num	sc_format_num
sc_get_language	sc_format_num_region	sc_format_num_region
sc_get_regional	sc_get_language	sc_get_language
sc_get_theme	sc_get_regional	sc_get_regional
sc_include	sc_get_theme	sc_get_theme
sc_include_library	sc_include	sc_include
sc_lookup	sc_include_library	sc_include_library
sc_reset_connection_edit	sc_label	sc_label
sc_reset_connection_new	sc_lookup	sc_lookup
sc_reset_global	sc_reset_connection_edit	sc_reset_connection_edit
sc_rollback_trans	sc_reset_connection_new	sc_reset_connection_new
sc_select	sc_reset_global	sc_reset_global
sc_set_fetchmode	sc_rollback_trans	sc_rollback_trans
sc_set_global	sc_select	sc_select
sc_sql_injection	sc_set_fetchmode	sc_set_fetchmode
sc_time_diff	sc_set_global	sc_set_global
sc_trunc_num	sc_sql_injection	sc_sql_injection
sc_url_library	sc_time_diff	sc_time_diff
sc_warning	sc_trunc_num	sc_trunc_num
sc_webservice	sc_url_library	sc_url_library
Variables - Database	sc_warning	sc_warning
	sc_webservice	sc_webservice
	Variables - Databases	Variables - Database

Onchange

Aplicação Calendário	Aplicação Formulário	Aplicação Controle
sc_ajax_javascript	sc_ajax_javascript	sc_ajax_javascript
sc_ajax_message	sc_ajax_message	sc_ajax_message
sc_begin_trans	sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_commit_trans	sc_commit_trans
sc_connection_edit	sc_connection_edit	sc_concat
sc_connection_new	sc_connection_new	sc_connection_edit
sc_date_conv	sc_date_conv	sc_connection_new
sc_date_dif	sc_date_dif	sc_date
sc_date_dif_2	sc_date_dif_2	sc_date_conv
sc_date_empty	sc_date_empty	sc_date_dif
sc_decode	sc_decode	sc_date_dif_2
sc_encode	sc_encode	sc_date_empty
sc_error_exit	sc_error_exit	sc_decode
sc_error_message	sc_error_message	sc_encode
sc_exec_sql	sc_exec_sql	sc_error_exit
sc_field_display	sc_field_display	sc_error_message
sc_field_readonly	sc_field_readonly	sc_exec_sql
sc_format_num	sc_format_num	sc_field_display
sc_get_language	sc_format_num_region	sc_field_readonly
sc_get_regional	sc_get_language	sc_format_num
sc_get_theme	sc_get_regional	sc_format_num_region
sc_include	sc_get_theme	sc_get_language
sc_include_library	sc_include	sc_get_regional
sc_lookup	sc_include_library	sc_get_theme
sc_reset_connection_edit	sc_label	sc_include
sc_reset_connection_new	sc_lookup	sc_include_library
sc_reset_global	sc_reset_connection_edit	sc_label
sc_rollback_trans	sc_reset_connection_new	sc_lookup
sc_select	sc_reset_global	sc_reset_connection_edit
sc_set_fetchmode	sc_rollback_trans	sc_reset_connection_new
sc_set_global	sc_select	sc_reset_global
sc_sql_injection	sc_set_fetchmode	sc_rollback_trans
sc_time_diff	sc_set_global	sc_select
sc_trunc_num	sc_sql_injection	sc_set_fetchmode
sc_url_library	sc_time_diff	sc_set_global
sc_warning	sc_trunc_num	sc_sql_injection
sc_webservice	sc_url_library	sc_time_diff
Variables - Database	sc_warning	sc_trunc_num
	sc_webservice	sc_url_library
	Variables - Database	sc_warning
		sc_webservice
		Variables - Database

OnClick

Aplicação Calendário	Aplicação Consulta	Aplicação Formulário	Aplicação Controle
sc_ajax_javascript	sc_ajax_refresh sc_begin_trans	sc_ajax_javascript	sc_ajax_javascript
sc_ajax_message	sc_calc_dv	sc_ajax_message	sc_ajax_message
sc_begin_trans	sc_commit_trans	sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_concat	sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_date	sc_commit_trans	sc_commit_trans
sc_concat	sc_date_conv	sc_concat	sc_concat
sc_connection_edit	sc_date_dif	sc_connection_edit	sc_connection_edit
sc_connection_new	sc_date_dif_2	sc_connection_new	sc_connection_new
sc_date	sc_date_empty	sc_date	sc_date
sc_date_conv	sc_decode	sc_date_conv	sc_date_conv
sc_date_dif	sc_encode	sc_date_dif	sc_date_dif
sc_date_dif_2	sc_exec_sql	sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_format_num	sc_date_empty	sc_date_empty
sc_decode	sc_format_num_region	sc_decode	sc_decode
sc_encode	sc_label	sc_encode	sc_encode
sc_error_exit	sc_lin_cod_barra_arrecadacao	sc_error_exit	sc_error_exit
sc_error_message	sc_lin_cod_barra_banco	sc_error_message	sc_error_message
sc_exec_sql	sc_lin_digitavel_arrecadacao	sc_exec_sql	sc_exec_sql
sc_field_display	sc_lin_digitavel_banco	sc_field_display	sc_field_display
sc_field_readonly	sc_lookup	sc_field_readonly	sc_field_readonly
sc_format_num	sc_mail_send	sc_format_num	sc_format_num
sc_get_language	sc_reset_global	sc_format_num_region	sc_format_num_region
sc_get_regional	sc_rollback_trans	sc_get_language	sc_get_language
sc_get_theme	sc_select	sc_get_regional	sc_get_regional
sc_include	sc_set_fetchmode	sc_get_theme	sc_get_theme
sc_include_library	sc_set_global	sc_include	sc_include
sc_lookup	sc_sql_injection	sc_include_library	sc_include_library
sc_reset_connection_edit	sc_sql_protect	sc_label	sc_label
sc_reset_connection_new	sc_time_diff	sc_lookup	sc_lookup
sc_reset_global	sc_trunc_num	sc_reset_connection_edit	sc_reset_connection_edit
sc_rollback_trans	sc_vl_extenso	sc_reset_connection_new	sc_reset_connection_new
sc_select	sc_where_current	sc_reset_global	sc_reset_global
sc_set_fetchmode	sc_where_filter	sc_rollback_trans	sc_rollback_trans
sc_set_focus	sc_where_orig	sc_select	sc_select
sc_set_global		sc_set_fetchmode	sc_set_fetchmode
sc_sql_injection		sc_set_focus	sc_set_global
sc_time_diff		sc_set_global	sc_sql_injection
sc_trunc_num		sc_sql_injection	sc_time_diff
sc_url_library		sc_time_diff	sc_trunc_num
sc_warning		sc_trunc_num	sc_url_library
sc_webservice		sc_url_library	sc_warning
Variables - Database		sc_warning	sc_webservice
		sc_webservice	Variables - Database
		Variables - Database	

OnFocus

Aplicação Calendário	Aplicação Formulário	Aplicação Controle
sc_ajax_javascript	sc_ajax_javascript	sc_ajax_javascript
sc_ajax_message	sc_ajax_message	sc_ajax_message
sc_begin_trans	sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_commit_trans	sc_commit_trans
sc_concat	sc_concat	sc_concat
sc_connection_edit	sc_connection_edit	sc_connection_edit
sc_connection_new	sc_connection_new	sc_connection_new
sc_date	sc_date	sc_date
sc_date_conv	sc_date_conv	sc_date_conv
sc_date_dif	sc_date_dif	sc_date_dif
sc_date_dif_2	sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_date_empty	sc_date_empty
sc_decode	sc_decode	sc_decode
sc_encode	sc_encode	sc_encode
sc_error_exit	sc_error_exit	sc_error_exit
sc_error_message	sc_error_message	sc_error_message
sc_exec_sql	sc_exec_sql	sc_exec_sql
sc_field_display	sc_field_display	sc_field_display
sc_field_readonly	sc_field_readonly	sc_field_readonly
sc_format_num	sc_format_num	sc_format_num
sc_get_language	sc_format_num_region	sc_format_num_region
sc_get_regional	sc_get_language	sc_get_language
sc_get_theme	sc_get_regional	sc_get_regional
sc_include	sc_get_theme	sc_get_theme
sc_include_library	sc_include	sc_include
sc_lookup	sc_include_library	sc_include_library
sc_reset_connection_edit	sc_label	sc_label
sc_reset_connection_new	sc_lookup	sc_lookup
sc_reset_global	sc_reset_connection_edit	sc_reset_connection_edit
sc_rollback_trans	sc_reset_connection_new	sc_reset_connection_new
sc_select	sc_reset_global	sc_reset_global
sc_set_fetchmode	sc_rollback_trans	sc_rollback_trans
sc_set_global	sc_select	sc_select
sc_sql_injection	sc_set_fetchmode	sc_set_fetchmode
sc_time_diff	sc_set_global	sc_set_global
sc_trunc_num	sc_sql_injection	sc_sql_injection
sc_url_library	sc_time_diff	sc_time_diff
sc_warning	sc_trunc_num	sc_trunc_num
sc_webservice	sc_url_library	sc_url_library
Variables - Database	sc_warning	sc_warning
	sc_webservice	sc_webservice
	Variables - Database	

onAfterDelete

Aplicação Calendário	Aplicação Formulário

Aplicação Calendário	Aplicação Formulário
sc_ajax_javascript	sc_ajax_javascript
sc_begin_trans	sc_begin_trans
sc_btn_display	sc_btn_display
sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_commit_trans
sc_concat	sc_concat
sc_connection_edit	sc_connection_edit
sc_connection_new	sc_connection_new
sc_date	sc_date
sc_date_conv	sc_date_conv
sc_date_dif	sc_date_dif
sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_date_empty
sc_decode	sc_decode
sc_encode	sc_encode
sc_error_delete	sc_error_delete
sc_error_exit	sc_error_exit
sc_error_message	sc_error_message
sc_exec_sql	sc_exec_sql
sc_field_display	sc_field_display
sc_field_readonly	sc_field_readonly
sc_format_num	sc_format_num
sc_get_language	sc_format_num_region
sc_get_regional	sc_get_language
sc_get_theme	sc_get_regional
sc_include	sc_get_theme
sc_include_library	sc_include
sc_lookup	sc_include_library
sc_mail_send	sc_lookup
sc_master_value	sc_mail_send
sc_redir	sc_master_value
sc_reset_connection_edit	sc_redir
sc_reset_connection_new	sc_reset_connection_edit
sc_reset_global	sc_reset_connection_new
sc_rollback_trans	sc_reset_global
sc_select	sc_rollback_trans
sc_set_fetchmode	sc_select
sc_set_global	sc_set_fetchmode
sc_site_ssl	sc_set_global
sc_sql_injection	sc_site_ssl
sc_time_diff	sc_sql_injection
sc_trunc_num	sc_time_diff
sc_url_exit	sc_trunc_num
sc_url_library	sc_url_exit
sc_user_logout	sc_url_library
sc_warning	sc_user_logout
sc_webservice	sc_warning
sc_zip_file	sc_webservice
	sc_zip_file

Aplicação Calendário Aplicação Formulário

onAfterDeleteAll

Aplicação Formulário

Aplicação Formulário

sc_ajax_javascript
sc_begin_trans
sc_block_display
sc_btn_display
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_error_exit
sc_error_message
sc_exec_sql
sc_field_display
sc_field_readonly
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lookup
sc_mail_send
sc_master_value
sc_redir
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_site_ssl
sc_sql_injection
sc_sql_protect
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_zip_file

onAfterInsert

Aplicação Calendário	Aplicação Formulário
sc_ajax_javascript	sc_ajax_javascript
sc_begin_trans	sc_begin_trans
sc_block_display	sc_block_display
sc_btn_display	sc_btn_display
sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_commit_trans
sc_concat	sc_concat
sc_connection_edit	sc_connection_edit
sc_connection_new	sc_connection_new
sc_date	sc_date
sc_date_conv	sc_date_conv
sc_date_dif	sc_date_dif
sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_date_empty
sc_decode	sc_decode
sc_encode	sc_encode
sc_error_exit	sc_error_exit
sc_error_insert	sc_error_insert
sc_error_message	sc_error_message
sc_exec_sql	sc_exec_sql
sc_field_display	sc_field_display
sc_field_readonly	sc_field_readonly
sc_format_num	sc_format_num
sc_get_language	sc_format_num_region
sc_get_regional	sc_get_language
sc_get_theme	sc_get_regional
sc_include	sc_get_theme
sc_include_library	sc_include
sc_lookup	sc_include_library
sc_mail_send	sc_lookup
sc_master_value	sc_mail_send
sc_redir	sc_master_value
sc_reset_connection_edit	sc_redir
sc_reset_connection_new	sc_reset_connection_edit
sc_reset_global	sc_reset_connection_new
sc_rollback_trans	sc_reset_global
sc_select	sc_rollback_trans
sc_set_fetchmode	sc_select
sc_set_global	sc_set_fetchmode
sc_site_ssl	sc_set_global
sc_sql_injection	sc_site_ssl
sc_time_diff	sc_sql_injection
sc_trunc_num	sc_time_diff
sc_url_exit	sc_trunc_num
sc_url_library	sc_url_exit
sc_user_logout	sc_url_library
sc_warning	sc_user_logout
sc_webservice	sc_warning
sc_zip_file	sc_webservice

sc_zip_file

sc_webservice
sc_zip_file

onAfterInsertAll

Aplicação Formulário

sc_ajax_javascript
sc_begin_trans
sc_block_display
sc_btn_display
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_error_exit
sc_error_message
sc_exec_sql
sc_field_display
sc_field_readonly
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lookup
sc_mail_send
sc_master_value
sc_redir
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_site_ssl
sc_sql_injection
sc_sql_protect
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library

sc_user_logout
sc_warning
sc_webservice
sc_zip_file

Aplicação Formulário

onAfterUpdate

Aplicação Calendário	Aplicação Formulário
sc_ajax_javascript	sc_ajax_javascript
sc_begin_trans	sc_begin_trans
sc_block_display	sc_block_display
sc_btn_display	sc_btn_display
sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_commit_trans
sc_concat	sc_concat
sc_connection_edit	sc_connection_edit
sc_connection_new	sc_connection_new
sc_date	sc_date
sc_date_conv	sc_date_conv
sc_date_dif	sc_date_dif
sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_date_empty
sc_decode	sc_decode
sc_encode	sc_encode
sc_error_exit	sc_error_exit
sc_error_message	sc_error_message
sc_error_update	sc_error_update
sc_exec_sql	sc_exec_sql
sc_field_display	sc_field_display
sc_field_readonly	sc_field_readonly
sc_format_num	sc_format_num
sc_get_language	sc_format_num_region
sc_get_regional	sc_get_language
sc_get_theme	sc_get_regional
sc_include	sc_get_theme
sc_include_library	sc_include
sc_lookup	sc_include_library
sc_mail_send	sc_lookup
sc_master_value	sc_mail_send
sc_redir	sc_master_value
sc_reset_connection_edit	sc_redir
sc_reset_connection_new	sc_reset_connection_edit
sc_reset_global	sc_reset_connection_new
sc_rollback_trans	sc_reset_global
sc_select	sc_rollback_trans
sc_set_fetchmode	sc_select
sc_set_global	sc_set_fetchmode
sc_site_ssl	sc_set_global
sc_sql_injection	sc_site_ssl
sc_time_diff	sc_sql_injection
sc_trunc_num	sc_time_diff

Aplicação Calendário	Aplicação Formulário
sc_url_exit	sc_trunc_num
sc_url_library	sc_url_exit
sc_user_logout	sc_url_library
sc_warning	sc_user_logout
sc_webservice	sc_warning
sc_zip_file	sc_webservice
	sc_zip_file

onAfterUpdateAll

Aplicação Formulário
sc_ajax_javascript
sc_begin_trans
sc_block_display
sc_btn_display
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_error_exit
sc_error_message
sc_exec_sql
sc_field_display
sc_field_readonly
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lookup
sc_mail_send
sc_master_value
sc_redir
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_site_ssl
sc_sql_injection

sc_sql_injection
sc_sql_protect
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_zip_file

Aplicação Formulário

onApplicationInit

Aplicação Calendário	Aplicação Gráfico	Aplicação Consulta	Aplicação Filtro	Ap
sc_apl_conf	sc_apl_conf	sc_apl_conf	sc_apl_conf	sc
sc_apl_status	sc_apl_status	sc_apl_status	sc_apl_status	sc
sc_calc_dv	sc_calc_dv	sc_calc_dv	sc_calc_dv	sc
sc_connection_edit	sc_connection_edit	sc_change_connection	sc_connection_edit	sc
sc_connection_new	sc_connection_new	sc_date	sc_connection_new	sc
sc_date	sc_date	sc_date_conv	sc_date	sc
sc_date_dif	sc_date_dif	sc_date_dif	sc_date_dif	sc
sc_date_dif_2	sc_date_dif_2	sc_date_dif_2	sc_date_dif_2	sc
sc_decode	sc_decode	sc_date_empty	sc_decode	sc
sc_encode	sc_encode	sc_decode	sc_encode	sc
sc_format_num	sc_field_display	sc_encode	sc_format_num	sc
sc_get_language	sc_format_num	sc_field_init_off	sc_format_num_region	sc
sc_get_regional	sc_get_language	sc_format_num	sc_get_language	sc
sc_get_theme	sc_get_regional	sc_format_num_region	sc_get_regional	sc
sc_image	sc_get_theme	sc_redir	sc_get_theme	sc
sc_include	sc_hide_groupby_rule	sc_reset_apl_conf	sc_include	sc
sc_include_library	sc_image	sc_reset_apl_status	sc_include_library	sc
sc_language	sc_include	sc_reset_change_connection	sc_language	sc
sc_ldap_login	sc_include_library	sc_reset_global	sc_reset_apl_conf	sc
sc_ldap_logout	sc_language	sc_set_global	sc_reset_apl_status	sc
sc_ldap_search	sc_redir	sc_set_groupby_rule	sc_reset_connection_edit	sc
sc_master_value	sc_reset_apl_conf	sc_set_language	sc_reset_connection_new	sc
sc_redir	sc_reset_apl_status	sc_set_pdf_name	sc_reset_global	sc
sc_reset_apl_conf	sc_reset_change_connection	sc_set_regional	sc_set_global	sc
sc_reset_apl_status	sc_reset_connection_edit	sc_set_theme	sc_set_language	sc
sc_reset_change_connectio	sc_reset_connection_new	sc_time_diff	sc_set_regional	sc
n sc_reset_connection_edit	sc_reset_global	sc_trunc_num	sc_set_theme	sc
sc_reset_connection_new	sc_set_global	Variables - Authentication	sc_site_ssl	sc
sc_reset_global	sc_set_groupby_rule		sc_time_diff	sc
sc_set_global	sc_set_language		sc_trunc_num	sc
sc_set_language	sc_set_pdf_name		sc_url_exit	sc
sc_set_regional	sc_set_regional		sc_url_library	sc
sc_set_theme	sc_set_theme		sc_warning	sc
sc_site_ssl	sc_site_ssl		sc_webservice	sc
sc_time_diff	sc_time_diff		sc_zip_file	sc
sc_trunc_num	sc_trunc_num			sc
sc_url_exit	sc_url_exit			sc
sc_url_library	sc_url_library			sc

sc_warning	sc_warning			sc
sc_webservice	sc_webservice			sc
sc_zip_file	sc_zip_file			sc
Aplicação Calendário	**Aplicação Gráfico**	**Aplicação Consulta**	**Aplicação Filtro**	sc **Ap**

onBeforeDelete

Aplicação Calendário	Aplicação Formulário
sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_commit_trans
sc_concat	sc_concat
sc_connection_edit	sc_connection_edit
sc_connection_new	sc_connection_new
sc_date	sc_date
sc_date_conv	sc_date_conv
sc_date_dif	sc_date_dif
sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_date_empty
sc_decode	sc_decode
sc_encode	sc_encode
sc_error_continue	sc_error_continue
sc_error_exit	sc_error_exit
sc_error_message	sc_error_message
sc_exec_sql	sc_exec_sql
sc_format_num	sc_format_num
sc_get_language	sc_format_num_region
sc_get_regional	sc_get_language
sc_get_theme	sc_get_regional
sc_include	sc_get_theme
sc_include_library	sc_include
sc_lookup	sc_include_library
sc_mail_send	sc_lookup
sc_master_value	sc_mail_send
sc_redir	sc_master_value
sc_reset_connection_edit	sc_redir
sc_reset_connection_new	sc_reset_connection_edit
sc_reset_global	sc_reset_connection_new
sc_rollback_trans	sc_reset_global
sc_select	sc_rollback_trans
sc_set_fetchmode	sc_select
sc_set_global	sc_set_fetchmode
sc_site_ssl	sc_set_global
sc_sql_injection	sc_site_ssl
sc_time_diff	sc_sql_injection
sc_trunc_num	sc_time_diff
sc_url_exit	sc_trunc_num
sc_url_library	sc_url_exit
sc_user_logout	sc_url_library
sc_warning	sc_user_logout
sc_webservice	sc_warning
sc_zip_file	sc_webservice
	sc_zip_file

onBeforeDeleteAll

Aplicação Formulário

sc_begin_trans
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_error_exit
sc_error_message
sc_exec_sql
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lookup
sc_mail_send
sc_redir
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_site_ssl
sc_sql_injection
sc_sql_protect
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_zip_file

onBeforeInsert

Aplicação Calendário	Aplicação Formulário
sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_commit_trans
sc_concat	sc_concat
sc_connection_edit	sc_connection_edit
sc_connection_new	sc_connection_new
sc_date	sc_date
sc_date_conv	sc_date_conv
sc_date_dif	sc_date_dif
sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_date_empty
sc_decode	sc_decode
sc_encode	sc_encode
sc_error_continue	sc_error_continue
sc_error_exit	sc_error_exit
sc_error_message	sc_error_message
sc_exec_sql	sc_exec_sql
sc_format_num	sc_format_num
sc_get_language	sc_format_num_region
sc_get_regional	sc_get_language
sc_get_theme	sc_get_regional
sc_include	sc_get_theme
sc_include_library	sc_include
sc_lin_cod_barra_arrecadacao	sc_include_library
sc_lin_cod_barra_banco	sc_lin_cod_barra_arrecadacao
sc_lin_digitavel_arrecadacao	sc_lin_cod_barra_banco
sc_lin_digitavel_banco	sc_lin_digitavel_arrecadacao
sc_lookup	sc_lin_digitavel_banco
sc_mail_send	sc_lookup
sc_master_value	sc_mail_send
sc_redir	sc_master_value
sc_reset_connection_edit	sc_redir
sc_reset_connection_new	sc_reset_connection_edit
sc_reset_global	sc_reset_connection_new
sc_rollback_trans	sc_reset_global
sc_select	sc_rollback_trans
sc_set_fetchmode	sc_select
sc_set_global	sc_set_fetchmode
sc_site_ssl	sc_set_global
sc_sql_injection	sc_site_ssl
sc_time_diff	sc_sql_injection
sc_trunc_num	sc_time_diff
sc_url_exit	sc_trunc_num
sc_url_library	sc_url_exit
sc_user_logout	sc_url_library
sc_warning	sc_user_logout
sc_webservice	sc_warning
sc_zip_file	sc_webservice
	sc_zip_file

onBeforeInsertAll

Aplicação Formulário

sc_begin_trans
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_error_exit
sc_error_message
sc_exec_sql
sc_field_display
sc_field_readonly
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lookup
sc_mail_send
sc_redir
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_site_ssl
sc_sql_injection
sc_sql_protect
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_zip_file

onBeforeUpdate

Aplicação Calendário	Aplicação Formulário
Aplicação Calendário	Aplicação Formulário
sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_calc_dv
sc_changed	sc_changed
sc_commit_trans	sc_commit_trans
sc_concat	sc_concat
sc_connection_edit	sc_connection_edit
sc_connection_new	sc_connection_new
sc_date	sc_date
sc_date_conv	sc_date_conv
sc_date_dif	sc_date_dif
sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_date_empty
sc_decode	sc_decode
sc_encode	sc_encode
sc_error_continue	sc_error_continue
sc_error_exit	sc_error_exit
sc_error_message	sc_error_message
sc_exec_sql	sc_exec_sql
sc_format_num	sc_format_num
sc_get_language	sc_format_num_region
sc_get_regional	sc_get_language
sc_get_theme	sc_get_regional
sc_include	sc_get_theme
sc_include_library	sc_include
sc_lin_cod_barra_arrecadacao	sc_include_library
sc_lin_cod_barra_banco	sc_lin_cod_barra_arrecadacao
sc_lin_digitavel_arrecadacao	sc_lin_cod_barra_banco
sc_lin_digitavel_banco	sc_lin_digitavel_arrecadacao
sc_lookup	sc_lin_digitavel_banco
sc_mail_send	sc_lookup
sc_master_value	sc_mail_send
sc_redir	sc_master_value
sc_reset_connection_edit	sc_redir
sc_reset_connection_new	sc_reset_connection_edit
sc_reset_global	sc_reset_connection_new
sc_rollback_trans	sc_reset_global
sc_select	sc_rollback_trans
sc_set_fetchmode	sc_select
sc_set_global	sc_set_fetchmode
sc_site_ssl	sc_set_global
sc_sql_injection	sc_site_ssl
sc_time_diff	sc_sql_injection
sc_trunc_num	sc_time_diff
sc_url_exit	sc_trunc_num
sc_url_library	sc_url_exit
sc_user_logout	sc_url_library
sc_warning	sc_user_logout
sc_webservice	sc_warning
sc_zip_file	sc_webservice
	sc_zip_file

onBeforeUpdateAll

Aplicação Formulário

sc_begin_trans
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_error_exit
sc_error_message
sc_exec_sql
sc_field_display
sc_field_readonly
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lookup
sc_mail_send
sc_redir
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_site_ssl
sc_sql_injection
sc_sql_protect
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_zip_file

onExecute

Aplicação Blank	Aplicação Menu	Aplicação Menu árvore
sc_apl_conf	sc_apl_conf	sc_apl_conf
sc_apl_status	sc_begin_trans	sc_begin_trans
sc_begin_trans	sc_calc_dv	sc_calc_dv
sc_calc_dv	sc_commit_trans	sc_commit_trans
sc_change_connection	sc_concat	sc_concat
sc_commit_trans	sc_connection_edit	sc_connection_edit
sc_concat	sc_connection_new	sc_connection_new
sc_connection_edit	sc_date	sc_date
sc_connection_new	sc_date_dif	sc_date_dif
sc_date	sc_date_dif_2	sc_date_dif_2
sc_date_conv	sc_decode	sc_decode
sc_date_dif	sc_encode	sc_encode
sc_date_dif_2	sc_exec_sql	sc_exec_sql
sc_date_empty	sc_format_num	sc_format_num
sc_decode	sc_format_num_region	sc_format_num_region
sc_encode	sc_get_language	sc_get_language
sc_exec_sql	sc_get_regional	sc_get_regional
sc_format_num	sc_get_theme	sc_get_theme
sc_format_num_region	sc_include	sc_include
sc_get_language	sc_include_library	sc_include_library
sc_get_regional	sc_lookup	sc_lookup
sc_get_theme	sc_mail_send	sc_mail_send
sc_image	sc_menu_item	sc_menu_item
sc_include	sc_redir	sc_redir
sc_include_lib	sc_reset_connection_edit	sc_reset_connection_edit
sc_include_library	sc_reset_connection_new	sc_reset_connection_new
sc_language	sc_reset_global	sc_rollback_trans
sc_lin_cod_barra_arrecadacao	sc_rollback_trans	sc_script_name
sc_lin_cod_barra_banco	sc_script_name	sc_select
sc_lin_digitavel_arrecadacao	sc_select	sc_set_fetchmode
sc_lin_digitavel_banco	sc_set_fetchmode	sc_set_global
sc_log_add	sc_set_global	sc_sql_injection
sc_log_split	sc_sql_injection	sc_trunc_num
sc_lookup	sc_time_diff	sc_url_library
sc_mail_send	sc_trunc_num	sc_warning
sc_make_link	sc_url_library	sc_webservice
sc_redir	sc_warning	sc_zip_file
sc_reset_apl_conf	sc_webservice	Variables - Database
sc_reset_apl_status	sc_zip_file	
sc_reset_change_connection	Variables - Database	
sc_reset_connection_edit		
sc_reset_connection_new		
sc_reset_global		
sc_rollback_trans		
sc_select		
sc_set_fetchmode		
sc_set_global		
sc_set_language		
sc_set_regional		

sc_set_regional
sc_set_theme
sc_site_ssl
sc_sql_injection
sc_sql_protect
sc_time_diff
sc_trunc_num
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_zip_file
Variables - Database

Aplicação Blank	Aplicação Menu	Aplicação Menu árvore

onFilterInit

Aplicação Filtro

sc_apl_conf
sc_apl_status
sc_begin_trans
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_decode
sc_encode
sc_exec_sql
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_image
sc_include
sc_include_library
sc_language
sc_lookup
sc_mail_send
sc_redir
sc_reset_apl_conf
sc_reset_apl_status
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select

sc_set_fetchmode
sc_set_global
sc_set_language
sc_set_regional
sc_set_theme
sc_site_ssl
sc_sql_injection
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library
sc_warning
sc_webservice
sc_zip_file
Variables - Database

Aplicação Filtro

onFilterRefresh

Aplicação Filtro

sc_begin_trans
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_exec_sql
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lookup
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_sql_injection
sc_time_diff
sc_trunc_num
sc_url_library
sc_warning
sc_webservice
Variables - Database

onFilterSave

Aplicação Filtro

sc_begin_trans
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_decode
sc_encode
sc_exec_sql
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lookup
sc_mail_send
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_sql_injection
sc_time_diff
sc_trunc_num
sc_url_library
sc_warning
sc_webservice
sc_zip_file
Variables - Database

onFilterValidate

Aplicação Filtro

sc_begin_trans
sc_calc_dv
sc_commit_trans
sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_error_exit
sc_error_message
sc_exec_sql
sc_format_num
sc_format_num_region
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lookup
sc_mail_send
sc_redir
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_set_language
sc_set_regional
sc_set_theme
sc_sql_injection
sc_sql_protect
sc_time_diff
sc_trunc_num
sc_url_library
sc_warning
sc_webservice
sc_zip_file
Variables - Database

onFooter

Aplicação Gráfico	Aplicação Consulta	Aplicação ReportPDF
sc_begin_trans	sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_commit_trans	sc_commit_trans
sc_concat	sc_concat	sc_concat
sc_connection_edit	sc_date	sc_connection_edit
sc_connection_new	sc_date_conv	sc_connection_new
sc_date	sc_date_dif	sc_date
sc_date_conv	sc_date_dif_2	sc_date_conv
sc_date_dif	sc_date_empty	sc_date_dif
sc_date_dif_2	sc_format_num	sc_date_dif_2
sc_decode	sc_format_num_region	sc_decode
sc_encode	sc_get_language	sc_encode
sc_exec_sql	sc_get_regional	sc_exec_sql
sc_format_num	sc_get_theme	sc_format_num
sc_get_groupby_rule	sc_image	sc_get_language
sc_get_language	sc_lin_cod_barra_arrecadacao	sc_get_theme
sc_get_regional	sc_lin_cod_barra_banco	sc_lin_cod_barra_arrecadacao
sc_get_theme	sc_lin_digitavel_arrecadacao	sc_lin_cod_barra_banco
sc_include_library	sc_lin_digitavel_banco	sc_lin_digitavel_arrecadacao
sc_lin_cod_barra_arrecadacao	sc_lookup	sc_lin_digitavel_banco
sc_lin_cod_barra_banco	sc_rollback_trans	sc_lookup
sc_lin_digitavel_arrecadacao	sc_select	sc_mail_send
sc_lin_digitavel_banco	sc_set_fetchmode	sc_reset_connection_edit
sc_lookup	sc_set_global	sc_reset_connection_new
sc_mail_send	sc_sql_injection	sc_reset_global
sc_make_link	sc_sql_protect	sc_select
sc_reset_connection_edit	sc_time_diff	sc_set_fetchmode
sc_reset_connection_new	sc_trunc_num	sc_set_global
sc_reset_global	sc_vl_extenso	sc_sql_injection
sc_select	sc_where_current	sc_time_diff
sc_set_fetchmode	sc_where_filter	sc_trunc_num
sc_set_global	sc_where_orig	sc_warning
sc_sql_injection	Variables - Authentication	sc_where_current
sc_time_diff	Variables - Database	sc_where_filter
sc_trunc_num	Variables - Totalling	sc_where_orig
sc_url_library	Variables - Totalling (group by)	sc_zip_file
sc_warning		Variables - Database
sc_webservice		Variables - Totalling
sc_where_current		
sc_where_filter		
sc_where_orig		
sc_zip_file		
Variables - Database		
Variables - Totalling		

onGroupBy

Aplicação Consulta
sc_begin_trans
sc_calc_dv
sc_commit_trans
sc_concat
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_exec_sql
sc_format_num_region
sc_groupby_label
sc_lookup
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_sql_injection
sc_sql_protect
sc_time_diff
sc_trunc_num
sc_where_current
sc_where_filter
sc_where_orig
Variables - Authentication
Variables - Database
Variables - Totalling
Variables - Totalling (group by)

onHeader

Aplicação Gráfico	Aplicação Consulta	Aplicação ReportPDF
sc_begin_trans	sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_calc_dv	sc_calc_dv
sc_change_connection	sc_commit_trans	sc_change_connection
sc_commit_trans	sc_concat	sc_commit_trans
sc_concat	sc_date	sc_concat
sc_connection_edit	sc_date_conv	sc_connection_edit
sc_connection_new	sc_date_dif	sc_connection_new
sc_date	sc_date_dif_2	sc_date
sc_date_conv	sc_date_empty	sc_date_conv
sc_date_dif	sc_decode	sc_date_dif
sc_date_dif_2	sc_encode	sc_date_dif_2
sc_date_empty	sc_exec_sql	sc_date_empty

sc_decode
sc_encode
sc_exec_sql
sc_field_display
sc_format_num
sc_get_groupby_rule
sc_get_language
sc_get_regional
sc_get_theme
sc_include
sc_include_library
sc_lin_cod_barra_arrecadacao
sc_lin_cod_barra_banco
sc_lin_digitavel_arrecadacao
sc_lin_digitavel_banco
sc_lookup
sc_mail_send
sc_make_link
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_sql_injection
sc_time_diff
sc_trunc_num
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_where_current
sc_where_filter
sc_where_orig
sc_zip_file
Variables - Database
Variables - Totalling

onHeader

Aplicação Gráfico	Aplicação Consulta	Aplicação ReportPDF
sc_format_num	sc_format_num	sc_decode
sc_format_num_region	sc_format_num_region	sc_encode
sc_lin_cod_barra_arrecadacao	sc_lin_cod_barra_arrecadacao	sc_exec_sql
sc_lin_cod_barra_banco	sc_lin_cod_barra_banco	sc_field_disabled
sc_lin_digitavel_arrecadacao	sc_lin_digitavel_arrecadacao	sc_format_num
sc_lin_digitavel_banco	sc_lin_digitavel_banco	sc_get_language
sc_lookup	sc_lookup	sc_get_regional
sc_mail_send	sc_mail_send	sc_get_theme
sc_reset_global	sc_reset_global	sc_include
sc_rollback_trans	sc_rollback_trans	sc_lin_cod_barra_arrecadacao
sc_select	sc_select	sc_lin_cod_barra_banco
sc_select_order	sc_select_order	sc_lin_digitavel_arrecadacao
sc_select_where	sc_select_where	sc_lin_digitavel_banco
sc_set_fetchmode	sc_set_fetchmode	sc_lookup
sc_set_global	sc_set_global	sc_mail_send
sc_sql_injection	sc_sql_injection	sc_reset_connection_edit
sc_time_diff	sc_time_diff	sc_reset_connection_new
sc_trunc_num	sc_trunc_num	sc_reset_global
sc_vl_extenso	sc_vl_extenso	sc_rollback_trans
sc_where_current	sc_where_current	sc_select
sc_where_filter	sc_where_filter	sc_set_fetchmode
sc_where_orig	sc_where_orig	sc_set_global
Variables - Database	Variables - Database	sc_sql_injection
		sc_time_diff
		sc_trunc_num
		sc_user_logout
		sc_warning
		sc_where_current
		sc_where_filter
		sc_where_orig
		sc_zip_file
		Variables - Database
		Variables - Totalling

onScriptInit

Aplicação Calendário	Aplicação Gráfico	Aplicação Consulta	Aplicação Formulário
sc_apl_conf	sc_apl_conf	sc_apl_conf	sc_apl_conf
sc_apl_status	sc_apl_status	sc_apl_status	sc_apl_status
sc_begin_trans	sc_begin_trans	sc_begin_trans	sc_begin_trans
sc_block_display	sc_block_display	sc_block_display	sc_block_display
sc_btn_copy	sc_btn_display	sc_btn_display	sc_btn_copy
sc_btn_display	sc_calc_dv	sc_calc_dv	sc_btn_display
sc_calc_dv	sc_change_connection	sc_change_connection	sc_calc_dv
sc_change_connection	sc_commit_trans	sc_commit_trans	sc_change_connection
sc_commit_trans	sc_concat	sc_concat	sc_commit_trans

sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_exec_sql
sc_exit
sc_field_display
sc_field_readonly
sc_format_num
sc_form_show
sc_get_language
sc_get_regional
sc_get_theme
sc_image
sc_include
sc_include_library
sc_label
sc_language
sc_lookup
sc_mail_send
sc_master_value
sc_redir
sc_reset_apl_conf
sc_reset_apl_status
sc_reset_change_connection
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_set_language
sc_set_regional
sc_set_theme
sc_site_ssl
sc_sql_injection
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_zip_file

Aplicação Calendário

sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_decode
sc_encode
sc_exec_sql
sc_field_display
sc_format_num
sc_get_groupby_rule
sc_get_language
sc_get_regional
sc_get_theme
sc_image
sc_include
sc_include_library
sc_label
sc_language
sc_lin_cod_barra_arrecadacao
sc_lin_cod_barra_banco
sc_lin_digitavel_arrecadacao
sc_lin_digitavel_banco
sc_lookup
sc_mail_send
sc_make_link
sc_redir
sc_reset_apl_conf
sc_reset_apl_status
sc_reset_change_connection
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_select_field
sc_select_order
sc_select_where
sc_set_fetchmode
sc_set_global
sc_set_language
sc_set_regional
sc_set_theme
sc_site_ssl
sc_sql_injection
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_where_current
sc_where_filter

Aplicação Gráfico

sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_exec_sql
sc_field_color
sc_field_display
sc_format_num
sc_format_num_region
sc_get_groupby_rule
sc_get_language
sc_get_regional
sc_get_theme
sc_hide_groupby_rule
sc_image
sc_include
sc_include_lib
sc_include_library
sc_label
sc_language
sc_lin_cod_barra_arrecadacao
sc_lin_cod_barra_banco
sc_lin_digitavel_arrecadacao
sc_lin_digitavel_banco
sc_log_add
sc_log_split
sc_lookup
sc_mail_send
sc_make_link
sc_redir
sc_reset_apl_conf
sc_reset_apl_status
sc_reset_change_connection
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_select_field
sc_select_order
sc_select_where
sc_set_fetchmode
sc_set_global
sc_set_language
sc_set_regional
sc_set_theme
sc_site_ssl
sc_sql_injection
sc_sql_protect
sc_time_diff

Aplicação Consulta

sc_concat
sc_connection_edit
sc_connection_new
sc_date
sc_date_conv
sc_date_dif
sc_date_dif_2
sc_date_empty
sc_decode
sc_encode
sc_exec_sql
sc_exit
sc_field_display
sc_field_readonly
sc_format_num
sc_format_num_region
sc_form_show
sc_get_language
sc_get_regional
sc_get_theme
sc_image
sc_include
sc_include_library
sc_label
sc_language
sc_lookup
sc_mail_send
sc_master_value
sc_redir
sc_reset_apl_conf
sc_reset_apl_status
sc_reset_change_connection
sc_reset_connection_edit
sc_reset_connection_new
sc_reset_global
sc_rollback_trans
sc_select
sc_set_fetchmode
sc_set_global
sc_set_language
sc_set_regional
sc_set_theme
sc_site_ssl
sc_sql_injection
sc_time_diff
sc_trunc_num
sc_url_exit
sc_url_library
sc_user_logout
sc_warning
sc_webservice
sc_zip_file
Variables - Authentication

Aplicação Formulário

	sc_where_orig sc_zip_file Variables - Database	sc_trunc_num sc_url_exit sc_url_library sc_user_logout sc_warning sc_webservice sc_where_current sc_where_filter sc_where_orig sc_zip_file Variables - Authentication Variables - Database	
Aplicação Calendário	**Aplicação Gráfico**	**Aplicação Consulta**	**Aplicação Formulário**

onLoad

Aplicação Calendário	Aplicação Formulário	Aplicação Controle	Aplicação Menu
sc_begin_trans	sc_begin_trans	sc_alert	sc_apl_conf
sc_block_display	sc_block_display	sc_begin_trans	sc_apl_status
sc_btn_copy	sc_btn_copy	sc_block_display	sc_appmenu_add_item
sc_btn_display	sc_btn_display	sc_btn_display	sc_appmenu_create
sc_btn_new	sc_btn_new	sc_calc_dv	sc_appmenu_exist_item
sc_calc_dv	sc_calc_dv	sc_change_connection	sc_appmenu_remove_item
sc_change_connection	sc_change_connection	sc_commit_trans	sc_appmenu_reset
sc_commit_trans	sc_commit_trans	sc_concat	sc_appmenu_update_item
sc_concat	sc_concat	sc_connection_edit	sc_begin_trans
sc_connection_edit	sc_connection_edit	sc_connection_new	sc_btn_disable
sc_connection_new	sc_connection_new	sc_date	sc_calc_dv
sc_date	sc_date	sc_date_conv	sc_commit_trans
sc_date_conv	sc_date_conv	sc_date_dif	sc_concat
sc_date_dif	sc_date_dif	sc_date_dif_2	sc_connection_edit
sc_date_dif_2	sc_date_dif_2	sc_date_empty	sc_connection_new
sc_date_empty	sc_date_empty	sc_decode	sc_date
sc_decode	sc_decode	sc_encode	sc_date_dif
sc_encode	sc_encode	sc_error_exit	sc_date_dif_2
sc_exec_sql	sc_exec_sql	sc_error_message	sc_decode
sc_field_disabled	sc_field_disabled	sc_exec_sql	sc_encode
sc_field_display	sc_field_display	sc_field_disabled	sc_exec_sql
sc_field_readonly	sc_field_readonly	sc_field_display	sc_format_num
sc_format_num	sc_format_num	sc_field_readonly	sc_format_num_region
sc_form_show	sc_format_num_region	sc_format_num	sc_get_language
sc_get_language	sc_form_show	sc_format_num_region	sc_get_regional
sc_get_regional	sc_get_language	sc_form_show	sc_get_theme
sc_get_theme	sc_get_regional	sc_get_language	sc_image
sc_image	sc_get_theme	sc_get_regional	sc_include
sc_include	sc_image	sc_get_theme	sc_include_library
sc_include_library	sc_include	sc_image	sc_language
sc_label	sc_include_library	sc_include_library	sc_lookup
sc_lin_cod_barra_arrecadacao	sc_label	sc_label	sc_mail_send
sc_lin_cod_barra_banco	sc_lin_cod_barra_arrecadacao	sc_lookup	sc_menu_delete
sc_lin_digitavel_arrecadacao	sc_lin_cod_barra_banco	sc_mail_send	sc_menu_disable
sc_lin_digitavel_banco	sc_lin_digitavel_arrecadacao	sc_master_value	sc_menu_force_mobile

Aplicação Calendário	Aplicação Formulário	Aplicação Controle	Aplicação Menu
sc_lin_digitavel_banco	sc_lin_digitavel_arrecadacao	sc_master_value	sc_menu_force_mobile
sc_log_split	sc_lin_digitavel_banco	sc_redir	sc_redir
sc_lookup	sc_log_split	sc_reset_connection_edit	sc_reset_apl_conf
sc_mail_send	sc_lookup	sc_reset_connection_new	sc_reset_apl_status
sc_master_value	sc_mail_send	sc_reset_global	sc_reset_change_connection
sc_redir	sc_master_value	sc_rollback_trans	sc_reset_connection_edit
sc_reset_connection_edit	sc_redir	sc_select	sc_reset_connection_new
sc_reset_connection_new	sc_reset_connection_edit	sc_set_fetchmode	sc_reset_global
sc_reset_global	sc_reset_connection_new	sc_set_focus	sc_reset_menu_delete
sc_rollback_trans	sc_reset_global	sc_set_global	sc_reset_menu_disable
sc_select	sc_rollback_trans	sc_site_ssl	sc_rollback_trans
sc_set_fetchmode	sc_select	sc_sql_injection	sc_select
sc_set_focus	sc_set_fetchmode	sc_sql_protect	sc_set_fetchmode
sc_set_global	sc_set_focus	sc_time_diff	sc_set_global
sc_site_ssl	sc_set_global	sc_trunc_num	sc_set_language
sc_sql_injection	sc_site_ssl	sc_url_exit	sc_set_regional
sc_time_diff	sc_sql_injection	sc_url_library	sc_set_theme
sc_trunc_num	sc_time_diff	sc_user_logout	sc_site_ssl
sc_url_exit	sc_trunc_num	sc_warning	sc_sql_injection
sc_url_library	sc_url_exit	sc_webservice	sc_time_diff
sc_user_logout	sc_url_library	sc_zip_file	sc_trunc_num
sc_warning	sc_user_logout	Variables - Database	sc_url_library
sc_webservice	sc_warning		sc_warning
sc_zip_file	sc_webservice		sc_webservice
	sc_zip_file		sc_zip_file
			Variables - Database

onNavigate

Aplicação Consulta	Aplicação Formulário
sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_block_display
sc_commit_trans	sc_btn_display
sc_concat	sc_calc_dv
sc_date	sc_commit_trans
sc_date_conv	sc_concat
sc_date_dif	sc_connection_edit
sc_date_dif_2	sc_connection_new
sc_date_empty	sc_date
sc_decode	sc_date_conv
sc_encode	sc_date_dif
sc_exec_sql	sc_date_dif_2
sc_format_num	sc_date_empty
sc_format_num_region	sc_decode
sc_get_groupby_rule	sc_encode
sc_get_language	sc_error_exit
sc_get_regional	sc_error_message
sc_get_theme	sc_exec_sql
sc_include	sc_field_disabled
sc_label	sc_field_display
sc_lin_cod_barra_arrecadacao	sc_field_readonly

Aplicação Consulta	Aplicação Formulário
sc_lin_cod_barra_banco	sc_format_num
sc_lin_digitavel_arrecadacao	sc_format_num_region
sc_lin_digitavel_banco	sc_get_language
sc_lookup	sc_get_regional
sc_rollback_trans	sc_include
sc_select	sc_include_library
sc_set_fetchmode	sc_label
sc_sql_injection	sc_lin_cod_barra_arrecadacao
sc_sql_protect	sc_lin_cod_barra_banco
sc_time_diff	sc_lin_digitavel_arrecadacao
sc_trunc_num	sc_lin_digitavel_banco
sc_vl_extenso	sc_lookup
sc_where_current	sc_mail_send
sc_where_filter	sc_master_value
sc_where_orig	sc_reset_connection_edit
Variables - Database	sc_reset_connection_new
	sc_reset_global
	sc_rollback_trans
	sc_select
	sc_set_fetchmode
	sc_site_ssl
	sc_sql_injection
	sc_time_diff
	sc_trunc_num
	sc_url_exit
	sc_url_library
	sc_user_logout
	sc_warning
	sc_webservice
	sc_zip_file

onRecord

Aplicação Consulta	Aplicação Formulário	Aplicação ReportPDF
sc_begin_trans	sc_begin_trans	sc_begin_trans
sc_calc_dv	sc_block_display	sc_calc_dv
sc_commit_trans	sc_btn_display	sc_change_connection
sc_concat	sc_calc_dv	sc_commit_trans
sc_date	sc_change_connection	sc_concat
sc_date_conv	sc_commit_trans	sc_connection_edit
sc_date_dif	sc_concat	sc_connection_new
sc_date_dif_2	sc_connection_edit	sc_date
sc_date_empty	sc_connection_new	sc_date_conv
sc_decode	sc_date	sc_date_dif
sc_encode	sc_date_conv	sc_date_dif_2
sc_exec_sql	sc_date_dif	sc_date_empty
sc_field_color	sc_date_dif_2	sc_decode
sc_field_style	sc_date_empty	sc_encode
sc_format_num	sc_decode	sc_exec_sql
sc_format_num_region	sc_encode	sc_field_color
sc_get_groupby_rule	sc_exec_sql	sc_get_language

sc_get_groupby_rule	sc_exec_sql	sc_get_language
sc_get_language	sc_field_disabled_record	sc_get_regional
sc_get_regional	sc_field_display	sc_get_theme
sc_get_theme	sc_field_readonly	sc_include
sc_include	sc_format_num	sc_link
Aplicação Consulta	**Aplicação Formulário**	**Aplicação ReportPDF**
sc_include_library	sc_format_num_region	sc_lin_cod_barra_arrecadacao
sc_link	sc_get_language	sc_lin_cod_barra_banco
sc_lin_cod_barra_arrecadacao	sc_get_regional	sc_lin_digitavel_arrecadacao
sc_lin_cod_barra_banco	sc_get_theme	sc_lin_digitavel_banco
sc_lin_digitavel_arrecadacao	sc_image	sc_lookup
sc_lin_digitavel_banco	sc_include	sc_mail_send
sc_lookup	sc_include_library	sc_reset_connection_edit
sc_mail_send	sc_label	sc_reset_connection_new
sc_make_link	sc_lin_cod_barra_arrecadacao	sc_reset_global
sc_reset_global	sc_lin_cod_barra_banco	sc_rollback_trans
sc_rollback_trans	sc_lin_digitavel_arrecadacao	sc_select
sc_select	sc_lin_digitavel_banco	sc_set_fetchmode
sc_seq_register	sc_lookup	sc_set_global
sc_set_fetchmode	sc_mail_send	sc_sql_injection
sc_set_global	sc_master_value	sc_sql_protect
sc_sql_injection	sc_reset_connection_edit	sc_time_diff
sc_sql_protect	sc_reset_connection_new	sc_trunc_num
sc_text_style	sc_reset_global	sc_url_exit
sc_time_diff	sc_rollback_trans	sc_user_logout
sc_trunc_num	sc_select	sc_warning
sc_vl_extenso	sc_set_fetchmode	sc_where_current
sc_where_current	sc_set_global	sc_where_filter
sc_where_filter	sc_site_ssl	sc_where_orig
sc_where_orig	sc_sql_injection	sc_zip_file
Variables - Database	sc_time_diff	Variables - Database
Variables - Totalling	sc_trunc_num	Variables - Totalling
	sc_url_exit	
	sc_url_library	
	sc_user_logout	
	sc_warning	
	sc_webservice	
	sc_zip_file	

onRefresh

Aplicação Calendário	Aplicação Formulário	Aplicação Controle
sc_begin_trans	sc_begin_trans	sc_begin_trans
sc_block_display	sc_block_display	sc_block_display
sc_btn_display	sc_btn_display	sc_btn_display
sc_calc_dv	sc_calc_dv	sc_calc_dv
sc_commit_trans	sc_commit_trans	sc_changed
sc_concat	sc_concat	sc_commit_trans
sc_date	sc_date	sc_concat
sc_date_conv	sc_date_conv	sc_connection_edit
sc_date_dif	sc_date_dif	sc_connection_new
sc_date_dif_2	sc_date_dif_2	sc_date
sc_date_empty	sc_date_empty	sc_date_conv
sc_decode	sc_decode	sc_date_dif
sc_encode	sc_encode	sc_date_dif_2
sc_exec_sql	sc_exec_sql	sc_date_empty
sc_field_display	sc_field_display	sc_decode
sc_field_readonly	sc_field_readonly	sc_encode
sc_format_num	sc_format_num	sc_error_exit
sc_get_language	sc_format_num_region	sc_error_message
sc_get_regional	sc_get_language	sc_exec_sql
sc_get_theme	sc_get_regional	sc_field_display
sc_include	sc_get_theme	sc_field_readonly
sc_include_library	sc_include	sc_format_num
sc_label	sc_include_library	sc_format_num_region
sc_lin_cod_barra_arrecadacao	sc_label	sc_get_language
sc_lin_cod_barra_banco	sc_lin_cod_barra_arrecadacao	sc_get_regional
sc_lin_digitavel_arrecadacao	sc_lin_cod_barra_banco	sc_get_theme
sc_lin_digitavel_banco	sc_lin_digitavel_arrecadacao	sc_include
sc_lookup	sc_lin_digitavel_banco	sc_include_library
sc_mail_send	sc_lookup	sc_lookup
sc_master_value	sc_mail_send	sc_mail_send
sc_reset_global	sc_master_value	sc_master_value
sc_rollback_trans	sc_reset_global	sc_redir
sc_select	sc_rollback_trans	sc_reset_connection_edit
sc_set_fetchmode	sc_select	sc_reset_connection_new
sc_set_focus	sc_set_fetchmode	sc_rollback_trans
sc_set_global	sc_set_focus	sc_select
sc_site_ssl	sc_set_global	sc_set_fetchmode
sc_sql_injection	sc_site_ssl	sc_set_focus
sc_time_diff	sc_sql_injection	sc_set_global
sc_trunc_num	sc_time_diff	sc_sql_injection
sc_url_exit	sc_trunc_num	sc_time_diff
sc_url_library	sc_url_exit	sc_trunc_num
sc_warning	sc_url_library	sc_url_library
sc_webservice	sc_warning	sc_warning
sc_zip_file	sc_webservice	sc_webservice
	sc_zip_file	sc_zip_file
		Variables - Database

onValidate

Aplicação Calendário	Aplicação Formulário	Aplicação Controle
sc_ajax_javascript	sc_ajax_javascript	sc_alert
sc_begin_trans	sc_begin_trans	sc_apl_conf
sc_block_display	sc_block_display	sc_apl_status
sc_btn_delete	sc_btn_delete	sc_begin_trans
sc_btn_display	sc_btn_display	sc_btn_display
sc_btn_insert	sc_btn_insert	sc_calc_dv
sc_btn_update	sc_btn_update	sc_commit_trans
sc_calc_dv	sc_calc_dv	sc_concat
sc_changed	sc_changed	sc_connection_edit
sc_commit_trans	sc_commit_trans	sc_connection_new
sc_concat	sc_concat	sc_date
sc_date	sc_date	sc_date_conv
sc_date_conv	sc_date_conv	sc_date_dif
sc_date_dif	sc_date_dif	sc_date_dif_2
sc_date_dif_2	sc_date_dif_2	sc_decode
sc_date_empty	sc_date_empty	sc_encode
sc_decode	sc_decode	sc_error_exit
sc_encode	sc_encode	sc_error_message
sc_error_exit	sc_error_exit	sc_exec_sql
sc_error_message	sc_error_message	sc_field_display
sc_exec_sql	sc_exec_sql	sc_field_readonly
sc_field_display	sc_field_display	sc_format_num_region
sc_field_readonly	sc_field_readonly	sc_get_theme
sc_format_num	sc_format_num	sc_include
sc_form_show	sc_format_num_region	sc_include_library
sc_get_language	sc_form_show	sc_language
sc_get_regional	sc_get_language	sc_ldap_login
sc_get_theme	sc_get_regional	sc_ldap_logout
sc_image	sc_get_theme	sc_ldap_search
sc_include	sc_image	sc_lin_cod_barra_arrecadacao
sc_include_library	sc_include	sc_lin_cod_barra_banco
sc_ldap_login	sc_include_library	sc_lin_digitavel_arrecadacao
sc_ldap_logout	sc_ldap_login	sc_lin_digitavel_banco
sc_ldap_search	sc_ldap_logout	sc_lookup
sc_log_add	sc_ldap_search	sc_mail_send
sc_lookup	sc_log_add	sc_make_link
sc_mail_send	sc_lookup	sc_redir
sc_master_value	sc_mail_send	sc_reset_apl_conf
sc_redir	sc_master_value	sc_reset_apl_status
sc_reset_global	sc_redir	sc_reset_change_connection
sc_rollback_trans	sc_reset_global	sc_reset_connection_edit
sc_select	sc_rollback_trans	sc_reset_connection_new
sc_set_fetchmode	sc_select	sc_reset_global
sc_set_global	sc_set_fetchmode	sc_rollback_trans
sc_site_ssl	sc_set_global	sc_select
sc_sql_injection	sc_site_ssl	sc_set_fetchmode
sc_sql_protect	sc_sql_injection	sc_set_global
sc_time_diff	sc_sql_protect	sc_set_language
sc_trunc_num	sc_time_diff	sc_set_regional

sc_url_exit	sc_trunc_num	sc_set_theme
sc_url_library	sc_url_exit	sc_site_ssl
sc_user_logout	sc_url_library	sc_sql_injection
sc_warning	sc_user_logout	sc_sql_protect
sc_webservice	sc_warning	sc_time_diff
sc_zip_file	sc_webservice	sc_trunc_num
	sc_zip_file	sc_url_exit
		sc_url_library
		sc_user_logout
		sc_warning
		sc_webservice
		sc_zip_file
		Variables - Database

onValidateFailure

Aplicação Calendário	Aplicação Formulário	Aplicação Controle
sc_ajax_javascript	sc_ajax_javascript	sc_ajax_javascript
sc_begin_trans	sc_begin_trans	sc_apl_conf
sc_block_display	sc_block_display	sc_apl_status
sc_btn_delete	sc_btn_delete	sc_begin_trans
sc_btn_display	sc_btn_display	sc_btn_display
sc_btn_insert	sc_btn_insert	sc_calc_dv
sc_btn_update	sc_btn_update	sc_commit_trans
sc_calc_dv	sc_calc_dv	sc_concat
sc_changed	sc_changed	sc_connection_edit
sc_commit_trans	sc_commit_trans	sc_connection_new
sc_date	sc_date	sc_date
sc_date_conv	sc_date_conv	sc_date_conv
sc_date_dif	sc_date_dif	sc_date_dif
sc_date_dif_2	sc_date_dif_2	sc_date_dif_2
sc_date_empty	sc_date_empty	sc_decode
sc_decode	sc_decode	sc_encode
sc_encode	sc_encode	sc_error_exit
sc_error_exit	sc_error_exit	sc_error_message
sc_error_message	sc_error_message	sc_exec_sql
sc_exec_sql	sc_exec_sql	sc_field_display
sc_field_display	sc_field_display	sc_field_readonly
sc_field_readonly	sc_field_readonly	sc_format_num
sc_format_num	sc_format_num	sc_format_num_region
sc_get_language	sc_format_num_region	sc_get_language
sc_get_regional	sc_get_language	sc_get_regional
sc_get_theme	sc_get_regional	sc_get_theme
sc_include	sc_get_theme	sc_image
sc_include_library	sc_include	sc_include
sc_mail_send	sc_include_library	sc_include_library
sc_master_value	sc_mail_send	sc_lookup
sc_reset_global	sc_master_value	sc_mail_send
sc_rollback_trans	sc_reset_global	sc_redir
sc_select	sc_rollback_trans	sc_reset_connection_edit
sc_set_fetchmode	sc_select	sc_reset_connection_new

sc_set_global	sc_set_fetchmode	sc_reset_global
sc_site_ssl	sc_set_global	sc_rollback_trans
sc_sql_injection	sc_site_ssl	sc_select
sc_time_diff	sc_sql_injection	sc_set_fetchmode
sc_trunc_num	sc_time_diff	sc_set_global
sc_url_exit	sc_trunc_num	sc_site_ssl
sc_url_library	sc_url_exit	sc_sql_injection
sc_warning	sc_url_library	sc_time_diff
sc_webservice	sc_warning	sc_trunc_num
sc_zip_file	sc_webservice	sc_url_exit
	sc_zip_file	sc_url_library
		sc_warning
		sc_webservice
		sc_zip_file
		Variables - Database

onValidateSuccess

Aplicação Calendário	Aplicação Formulário	Aplicação Controle
sc_ajax_javascript	sc_ajax_javascript	sc_apl_conf
sc_begin_trans	sc_begin_trans	sc_apl_status
sc_block_display	sc_block_display	sc_begin_trans
sc_btn_delete	sc_btn_delete	sc_calc_dv
sc_btn_display	sc_btn_display	sc_commit_trans
sc_btn_insert	sc_btn_insert	sc_concat
sc_btn_update	sc_btn_update	sc_connection_edit
sc_calc_dv	sc_calc_dv	sc_connection_new
sc_changed	sc_changed	sc_date
sc_commit_trans	sc_commit_trans	sc_date_conv
sc_concat	sc_concat	sc_date_dif
sc_connection_edit	sc_connection_edit	sc_date_dif_2
sc_connection_new	sc_connection_new	sc_decode
sc_date	sc_date	sc_encode
sc_date_conv	sc_date_conv	sc_error_exit
sc_date_dif	sc_date_dif	sc_error_message
sc_date_dif_2	sc_date_dif_2	sc_exec_sql
sc_date_empty	sc_date_empty	sc_format_num
sc_decode	sc_decode	sc_format_num_region
sc_encode	sc_encode	sc_get_language
sc_error_exit	sc_error_exit	sc_get_regional
sc_error_message	sc_error_message	sc_get_theme
sc_exec_sql	sc_exec_sql	sc_image
sc_field_display	sc_field_display	sc_include
sc_field_readonly	sc_field_readonly	sc_include_library
sc_format_num	sc_format_num	sc_language
sc_get_language	sc_format_num_region	sc_lin_cod_barra_arrecadacao
sc_get_regional	sc_get_language	sc_lin_cod_barra_banco
sc_get_theme	sc_get_regional	sc_lin_digitavel_arrecadacao
sc_include	sc_get_theme	sc_lin_digitavel_banco
sc_include_library	sc_include	sc_lookup
sc_lin_cod_barra_arrecadacao	sc_include_library	sc_mail_send
sc_lin_cod_barra_banco	sc_lin_cod_barra_arrecadacao	sc_make_link

sc_lin_cod_barra_banco	sc_lin_cod_barra_arrecadacao	sc_make_link
sc_lin_digitavel_arrecadacao	sc_lin_cod_barra_banco	sc_redir
sc_lin_digitavel_banco	sc_lin_digitavel_arrecadacao	sc_reset_apl_conf
sc_lookup	sc_lin_digitavel_banco	sc_reset_apl_status
sc_mail_send	sc_lookup	sc_reset_change_connection
Aplicação Calendário	**Aplicação Formulário**	**Aplicação Controle**
sc_redir	sc_mail_send	sc_reset_connection_edit
sc_reset_connection_edit	sc_redir	sc_reset_connection_new
sc_reset_connection_new	sc_reset_connection_edit	sc_reset_global
sc_reset_global	sc_reset_connection_new	sc_rollback_trans
sc_rollback_trans	sc_reset_global	sc_select
sc_select	sc_rollback_trans	sc_set_fetchmode
sc_set_fetchmode	sc_select	sc_set_global
sc_set_focus	sc_set_fetchmode	sc_set_language
sc_set_global	sc_set_focus	sc_set_regional
sc_site_ssl	sc_set_global	sc_set_theme
sc_sql_injection	sc_site_ssl	sc_site_ssl
sc_time_diff	sc_sql_injection	sc_sql_injection
sc_trunc_num	sc_time_diff	sc_time_diff
sc_url_exit	sc_trunc_num	sc_trunc_num
sc_url_library	sc_url_exit	sc_url_exit
sc_user_logout	sc_url_library	sc_url_library
sc_warning	sc_user_logout	sc_user_logout
sc_webservice	sc_warning	sc_warning
sc_zip_file	sc_webservice	sc_webservice
	sc_zip_file	sc_zip_file
		Variables - Database