

Status: Validated

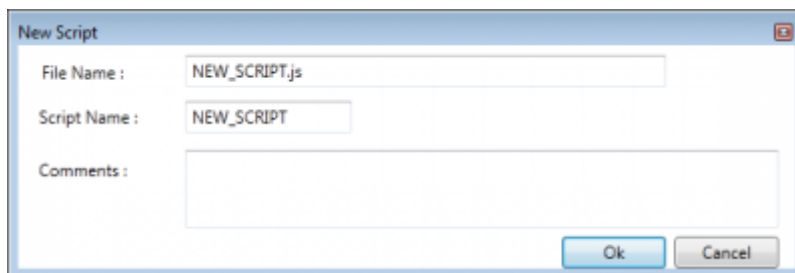
20/10/2016	spavel	Validated
05/10/2016	plong	Updated

Script Management

Add a New Script

To add a new script, right-click outside any node of the [Script Tree](#) and select the **'Add a new Script'** command.

The following dialog lets you enter:



- The script file name,
- The script name that will be displayed in Studio,
- The script comments.

Click the **'Ok'** button to validate your entries.

The new script file is physically created in the **'local'** subdirectory of the project (see [Project Structure and Organization](#)).

Notes:

- To enable **Breakpoints** to be inserted in the new script, it **MUST** have a unique name,
- If an existing script file name is used , the corresponding script is included in the project. The file content remains unchanged,
- By default, each new script is included in the **GLOBAL Process**.
If you wish to include a new script in **another application**, right-click the application node in the [Project Tree](#) and select the **'Add a new Script'** command.
This is required for backwards compatibility; there is no particular advantage in including a script in an application rather than in the GLOBAL Process.

Edit a Script

To edit a script, **double-click** its node in the [Script Tree](#).

If the script is already opened in a Code Editor instance, this instance is brought to the front. Otherwise, a new Code Editor instance is opened with the selected script in Editing mode.

Double-clicking a MapItem node of the script (such as Function or Scenario):

- Opens the script in Editing mode,
- Automatically navigates within the script to select and display the corresponding code block.

Notes:

- Editing a **Framework** or **Library** script requires special rights. These special scripts are in **read-only mode**.

Move a Script in a Folder

To add a script to a new folder, right-click the script node in the [Script Tree](#) and select the '**Create Folder**' command.

Desktop Studio creates a 'new folder' and moves the selected script to it.

You can rename the folder by selecting its node and hitting the **F2** key. Enter the new folder name, and hit **Enter**.

To move a script into an existing folder, simply **drag & drop it** onto the folder node in the [Script Tree](#).

To move a script out of its containing folder, simply **drag & drop it** out of the folder node in the [Script Tree](#).

Notes:

- It is impossible **to move a folder in another folder**: Desktop Studio only manages single-level folders,
- The folder is **not physically** created on the disk: it is only a 'logical' folder.

Rename a Script

You can rename a script by selecting its node in the [Script Tree](#) and hitting the **F2** key. Enter the new script name, and hit **Enter**.

Notes:

- The action of changing the name only modifies the logical name in Desktop Studio. The name of the physical file **containing the script is not affected by this operation.**
====Exclude a Script====

A Script can be temporary excluded from compilation and execution.

To exclude a script, right-click the script node in the [Script Tree](#) and select the '**Exclude Script**' command.

After confirmation, the script file is excluded from the project.

====Delete a Script====

To delete a script, right-click the script node in the [Script Tree](#) and select the '**Delete Script**' command.

After confirmation, the Script file is removed from the project.

Notes:

- Deleting a script **doesn't delete the script file on the disk.**

====Edit the XML List of Scripts====

If necessary (not recommended), it is possible to directly edit the XML code describing the list of scripts.

To do this, double-click the '**SCRIPTS**' node of the **GLOBAL** Process in the [Project Tree](#). The corresponding XML code opens in a new Code Editor instance:

```

1 <<SCRIPTS>
2 <!-- framework files -->
3 <SCRIPT Name="Constants" Src="%Sdk%\lib\common\ctx.enum.js" Folder="Framework" />
4 <SCRIPT Name="Ctx Core" Src="%Sdk%\lib\ctx\ctx.core.js" Folder="Framework" />
5 <SCRIPT Name="Ctx XML" Src="%Sdk%\lib\ctx\ctx.xml.js" Folder="Framework" />
6 <SCRIPT Name="Ctx Scenario" Src="%Sdk%\lib\ctx\ctx.scenario.js" Folder="Framework" />
7 <SCRIPT Name="Ctx Request" Src="%Sdk%\lib\ctx\ctx.request.js" Folder="Framework" />
8 <SCRIPT Name="Ctx Language" Src="%Sdk%\lib\ctx\ctx.language.js" Folder="Framework" />
9 <SCRIPT Name="Ctx Item" Src="%Sdk%\lib\ctx\ctx.item.js" Folder="Framework" />
10 <SCRIPT Name="Ctx Page" Src="%Sdk%\lib\ctx\ctx.page.js" Folder="Framework" />
11 <SCRIPT Name="Ctx Application" Src="%Sdk%\lib\ctx\ctx.application.js" Folder="Framework" />
12 <SCRIPT Name="Ctx Systray" Src="%Sdk%\lib\ctx\ctx.systray.js" Folder="Framework" />
13 <SCRIPT Name="Ctx Popup" Src="%Sdk%\lib\ctx\ctx.popup.js" Folder="Framework" />
14 <!-- generated definition file -->
15 <SCRIPT Name="Declarations" Src="BENCHMARK.resources.js" ReadOnly="Y" />
16 <!-- project files -->
17 <SCRIPT Name="Main Project" Src="BENCHMARK.js" />
18 <SCRIPT Name="FranFinance" Src="FranFinance.js" />
19 <SCRIPT Name="Orchestrator" Src="Orchestrator.js" />
20 <SCRIPT Name="Solfea" Src="Solfea.js" />
21 <SCRIPT Name="Sofinco" Src="Sofinco.js" />
22 <SCRIPT Name="Excel" Src="%Sdk%\lib\office\excel.js" Folder="Framework" />
23 <SCRIPT Name="Fso" Src="%Sdk%\lib\utils\fso.js" Folder="Framework" />
24 <SCRIPT Name="String" Src="%Sdk%\lib\utils\string.js" Folder="Framework" />
25 </SCRIPTS>

```

You can then edit the XML and save it. The [Script Tree](#) will be refreshed to reflect the modifications.

Notes:

- Please note: if this XML code remains open, it will not be synchronized with further modifications made in the [Project Tree](#). So, to avoid losing modifications, please close it as quickly as possible.

From: <https://contextor.eu/dokuwiki2/> - **SAP Intelligent Robotic Process Automation**

Permanent link: <https://contextor.eu/dokuwiki2/doku.php?id=guides:studiov3:scriptmanagement>

Last update: **2019/03/27 18:34**

