zenon manual

Scripts

v.7.11

Contents

1. Welcome to COPA-DATA help

GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com (mailto:documentation@copadata.com).

PROJECT SUPPORT

You can receive support for any real project you may have from our Support Team, who you can contact via email at support@copadata.com (mailto:support@copadata.com).

LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email sales@copadata.com (mailto:sales@copadata.com).

2. Scripts

Scripts make it possible to compile several user-defined functions and execute these in a sequence in Runtime. zenon provides some pre-defined scripts (on page 8).

License information

Part of the standard license of the Editor and Runtime.

3. Context menu and tool bar for scripts

CONTEXT MENU PROJECT MANAGER

Menu item	Action
`Script new`	Creates a new script.
`Export XML all ...`	Exports all entries as an XML file.
`Import XML...`	Imports entries from an XML file.
`Editor profile`	Opens the drop-down list with predefined editor profiles.
`Help`	Opens online help.

SCRIPTS TOOL BAR DETAIL VIEW

Parameters	Description
`Script new`	Adds a new script to the list.
`Add functions`	Opens the dialog for adding functions (on page 7).
`Use of scripts`	Opens the project analysis in the main window and displays the elements with which the selected script is used.
`Show unused scripts`	Opens the project analysis in the main window and shows unused scripts.
`Copy`	Copies the selected script or the selected functions to the Windows clipboard.
`Paste`	Pastes an element that was copied to the Windows clipboard.
`Delete`	Deletes the selected script or removes the selected function from the script.
`Export selected XML`	Exports selected entries as an XML file.
`Import XML`	Imports elements from an XML file.
`Edit selected cell`	Renames the script. It is also possible by left-clicking the field with the mouse or by pressing **`F2`**.
`Properties`	Opens the property window for the selected script.
`Help`	Opens online help

CONTEXT MENU SELECTED SCRIPT

TAG	Description
`Add functions`	Opens the dialog for adding functions (on page 7).
`Script new`	Adds a new script to the list.
`Create standard function`	Automatically creates a standard function for the selected script.
`Use of scripts`	Opens the project analysis in the main window and displays the elements with which the selected script is used.
`Show unused scripts`	Opens the project analysis in the main window and shows unused scripts.
`Copy`	Copies the selected script or the selected functions to the Windows clipboard.
`Paste`	Pastes an element that was copied to the Windows clipboard.

Delete	Deletes the selected script or removes the selected function from the script.
Export selected XML	Exports selected entries as an XML file.
Import XML	Imports elements from an XML file.
Edit selected cell	Renames the script. It is also possible by left-clicking the field with the mouse or by pressing **F2**.
Properties	Opens the property window for the selected script.
Help	Opens online help

4. Engineering in the Editor

To configure or edit scripts, click on the Scripts node and select the desired action in the tool bar or in the context menu.

> **Information**
>
> *When you change the script name, the parameters of the linked functions are changed accordingly. With multi-user projects, functions are changed automatically after requesting confirmation (Check out). After the name has been changed successfully, the script is changed.*

CONFIGURING A SCRIPT

To configure a new script:

1. Select **New symbol** in the context menu or in the tool bar.
2. A new script is added to the list.
3. Issue a name or select a pre-defined script (on page 8) from the drop-down list.
4. Add functions and sort them.
   Functions are executed in the sequence of the project configuration.
5. Create a function to start (on page 10) the script in Runtime.

ADD FUNCTIONS AND PLACE THEM IN SEQUENCE

You can add new functions to a script using a command or add or change the sequence of previously-linked functions by dragging & dropping.

FUNCTION NEW

To add a function to a script using a command:

1. Select **`Add functions`** in the context menu or in the tool bar.
2. The dialog to select one or several functions is opened.
3. Select the desired functions and close the dialog by clicking OK
4. The functions are added to the end of the list of pre-existing functions.

DRAG & DROP ACTIONS

To add a function to a script by dragging & dropping:

1. Select the desired function from the a script.
2. Move the function to the desired position:
   - Within the source script: The sequence of the function is changed.
     Hint: If the `Shift` key is held when dragging & dropping, the function is copied.
   - In a different script: The function is copied to the target script.
     If the node of the script is closed, the function is added at the end.
     If the node is opened, the function is inserted at the displayed location.
     Hint: If the `Shift` key is held when dragging & dropping, the function is deleted in the source script.

5. Pre-defined scripts

zenon provides a range of pre-defined scripts. The action of this script in Runtime is pre-defined. The functions to be executed must be configured individually.

Script	Description
AUTOSTART	The start information (e.g. open headings and status lines, reset alarm outputs etc.) is configured in the script. The script will be executed automatically at the start of Runtime when the start screen is called up if the project is the Runtime start project. The script is not executed, if the project is not defined as the start project or if it is a sub-project in a multi-hierarchical network. **Attention:** The start screen must not be executed in the AUTOSTART script because it is defined via the project property Start screen. For switching to a partial screens that belongs to a main screen, such as screen-specific menus etc., a separate script is defined. This ensures that on selection of the screen via the Screen switch function, selection from the screen catalog or the Return to last screen function, all detail screens are always opened with it.
AUTOEND	The defined end functions are configured in the script. The script is executed automatically when Runtime is ended if the project is the Runtime start project. The script is not executed, if the project is not defined as the start project or if it is a sub-project in a multi-hierarchical network. **Attention:** - The VBA Event **Project.Inactive** is executed by the **AUTOEND_xxx** script. Therefore the zenon function **Execute VBA macro** is no longer executed in scripts as VBA is not running at this time. VBA macros which should be carried out in **AUTOEND** must be called via **Project.Inactive**. - Cyclical archives must not be included in the **AUTOEND** script.
AUTOSTART_CLIENT	The start information (e.g. open headings and status lines, reset alarm outputs etc.) is configured in the script. The script will be executed automatically at start of the Runtime on a client and before activation of start display on a client if the project is the Runtime start project.
AUTOEND_CLIENT	The defined end functions are configured in the script. The script will be executed automatically at the end of Runtime on a client if the project is the Runtime start project.
AUTOSTART_SRVPRJ	Script will be executed automatically at the start of Runtime operation on the computer that is the server for the project, before opening the start screen,

	regardless of whether the project has been defined as the start project or whether it is a sub-project in a multi-hierarchical network.
AUTOEND_SRVPRJ	Script will be executed automatically at the end of Runtime operation on the computer that is the server for the project, regardless of whether the project has been defined as the start project or whether it is a sub-project in a multi-hierarchical network.

The following scripts are available for simulation mode; they are executed automatically:

Script	Description
AUTOSTART_SIMUL	If Runtime is started in simulation mode, the script with the name **AUTOSTART_SIMUL** is executed automatically if it is available.
AUTOEND_SIMUL	If Runtime is ended in simulation mode, the script with the name **AUTOEND_SIMUL** is executed automatically if it is available.

6. Executing scripts in Runtime

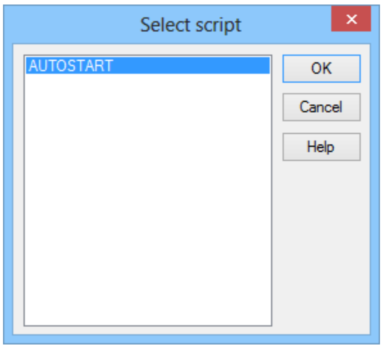
Scripts can be started in Runtime by means of functions: zenon provides two functions for this:

- **Script: Execute** Executes a script that was pre-defined in the Editor in Runtime.
- **Script: Select online** Allows selection of the script to be executed in Runtime.

To configure a function to execute a script:

1. Select the **New function** command in the **Functions** node
2. The dialog to select the function is opened
3. Navigate to the **Scripts** node
4. select the desired function:
   a) **Execute script** to execute a pre-defined script
   b) **Scripts with online selection** in order to make it possible to select a script which should be started in the Runtime.
5. link the function to a button

SELECT SCRIPT DIALOG



Parameters	Description
List of scripts	Shows all defined scripts. Selection via mouse click.
OK	Applies settings and closes the dialog. With **Execute script**, the selected script is started in Runtime when this function is executed. With **Script with online selection**, this dialog is opened in Runtime when this function is executed.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.

SCRIPTS IN THE NETWORK

At active network you can define via property node Execution for scripts on which computer the script should be executed.

Information

Take care of the priorities of the script execution in the network (on page 11)

7. Priorities in the network

At active network you define via proeprty node Execution for scripts where the script is executed:

- `Current computer`
- `Client`
- `Server`
- `Standby`

This execution location is true for functions contained in the script. Function which had a different execution location assigned, take over the execution location defined here.

EXAMPLE 1: CURRENT COMPUTER

- Function 1 was defined with execution location `Server`.
- Function 2 was defined with execution location `Client`.
- For function **`Execute script`** the `current computer` was defined.
- Consequence: Both functions are carried out on the same computer; the computer in which fucntion **`Execute scripts`** is executed. This can be a Client, a Server or a Standby.

EXAMPLE 2: CLIENT VERSUS SERVER

- The script contains 2 functions with execution location `Client`.
- For function **`Execute script`** `Server` was defined.
- Consequence: Both scripts are executed on the Server no matter on which computer **`Execute script`** was started.

EXAMPLE 3: DRIVER COMMANDS:

Driver commands are executed on the Server and the Standby per default. If a script with driver commands is started on the Client, then:

- the function driver command is started on the Client
- the driver is stopped after the execution of the script on the Server