zenon
by COPA-DATA

zenon manual

Smart Objects

v.8.20

Contents

1 Welcome to COPA-DATA help

ZENON VIDEO TUTORIALS

You can find practical examples for project configuration with zenon in our YouTube channel (https://www.copadata.com/tutorial_menu). The tutorials are grouped according to topics and give an initial insight into working with different zenon modules. All tutorials are available in English.

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.

PROJECT SUPPORT

You can receive support for any real project you may have from our customer service team, which you can contact via email at 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.

2 Smart Objects

The concept of Smart Objects is based on the idea of providing configuration contents for not so tech-savvy users of the zenon Editor in pre-configured objects known as smart objects. All the configuration options available in zenon are combined in small units. This offers the advantage that several encapsulated and separate objects within the zenon Editor can be maintained easily. They are completely reusable and make a zenon project scalable.

Two user roles are available:

- Project engineer of smart object templates

The creator of smart object templates is usually an experienced zenon user. He or she merges complex configurations in one package. In addition to the configurations performed such as the creation of symbols, the setting of interlocking conditions or the parameterization of variables and drivers, the complexity can be reduced for less experienced users. The creator of a smart object template can pre-select the properties available to the user of smart objects.

- User of smart objects

  Smart objects are based on a smart object template. The user can integrate the configurations merged in a smart object template into a zenon project quite easily. This therefore reduces the complexity because the zenon properties that need to be parameterized have already been pre-selected and restricted in the template.

3 Configuration - General notes

The **Smart Objects** module allows complex content to be merged in one package. These configurations can be exchanged between computers and zenon projects. A smart object template can be used any number of times in a project. The parameters of these instances, the smart objects, can be set differently.

The project engineer of a smart object template can release individual properties for parameterization. This makes the configuration of content in zenon quick, efficient and very easy.

Smart objects offer:

- Easy interchangeability of configurations via import and export.
- Simplified parameterization since only customizable properties are linked with the smart object.
- Dialog-based modification for updating existing smart objects and variable mapping.

The zenon Editor offers two user interfaces for configuration:

- The **Smart Object templates** tab for the configuration of smart objects.
- The **Smart Objects** node in the project tree for use and parameterization in zenon projects, modules and screens.

4 Configure and parametrize smart objects

Smart object templates are applied to an existing configuration in the project node of the zenon Editor. For this, smart objects are configured. A smart object is always based on a smart object template.

Information

You can find information on creating smart object templates in the Configuration of smart object templates (on page 45) chapter.

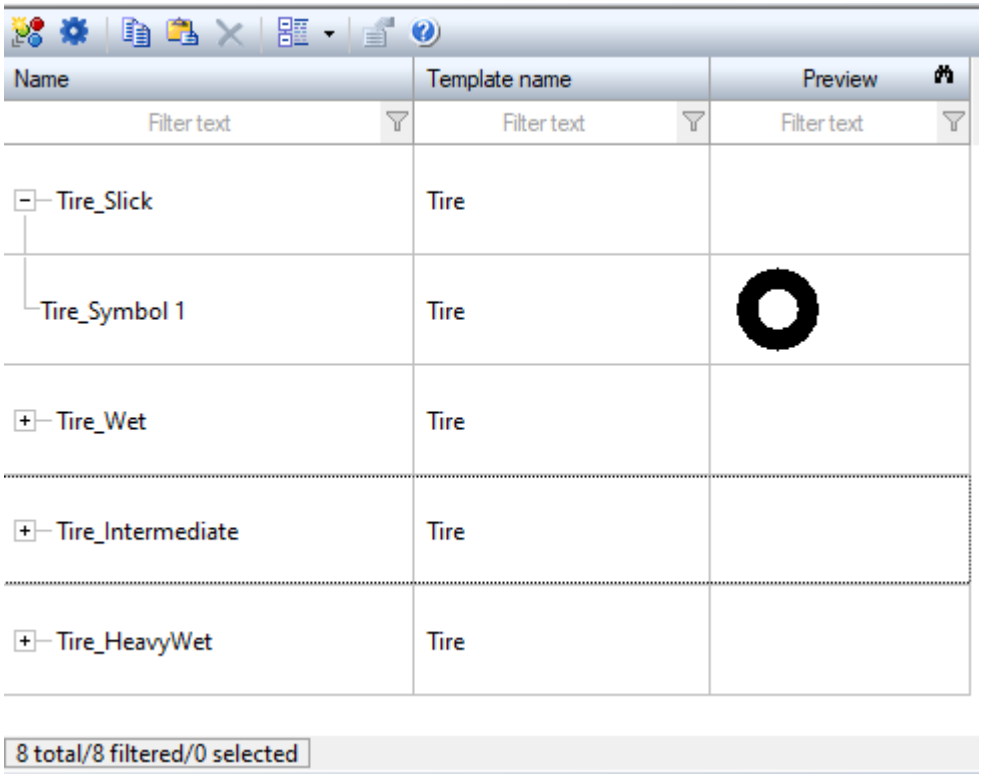
4.1 Context menu and toolbar

Smart objects can be managed using the toolbar and context menus.

TREE VIEW

Parameter	Description
Editor profile	Opens the drop-down list with predefined editor profiles.
Help	Opens online help.

DETAIL VIEW



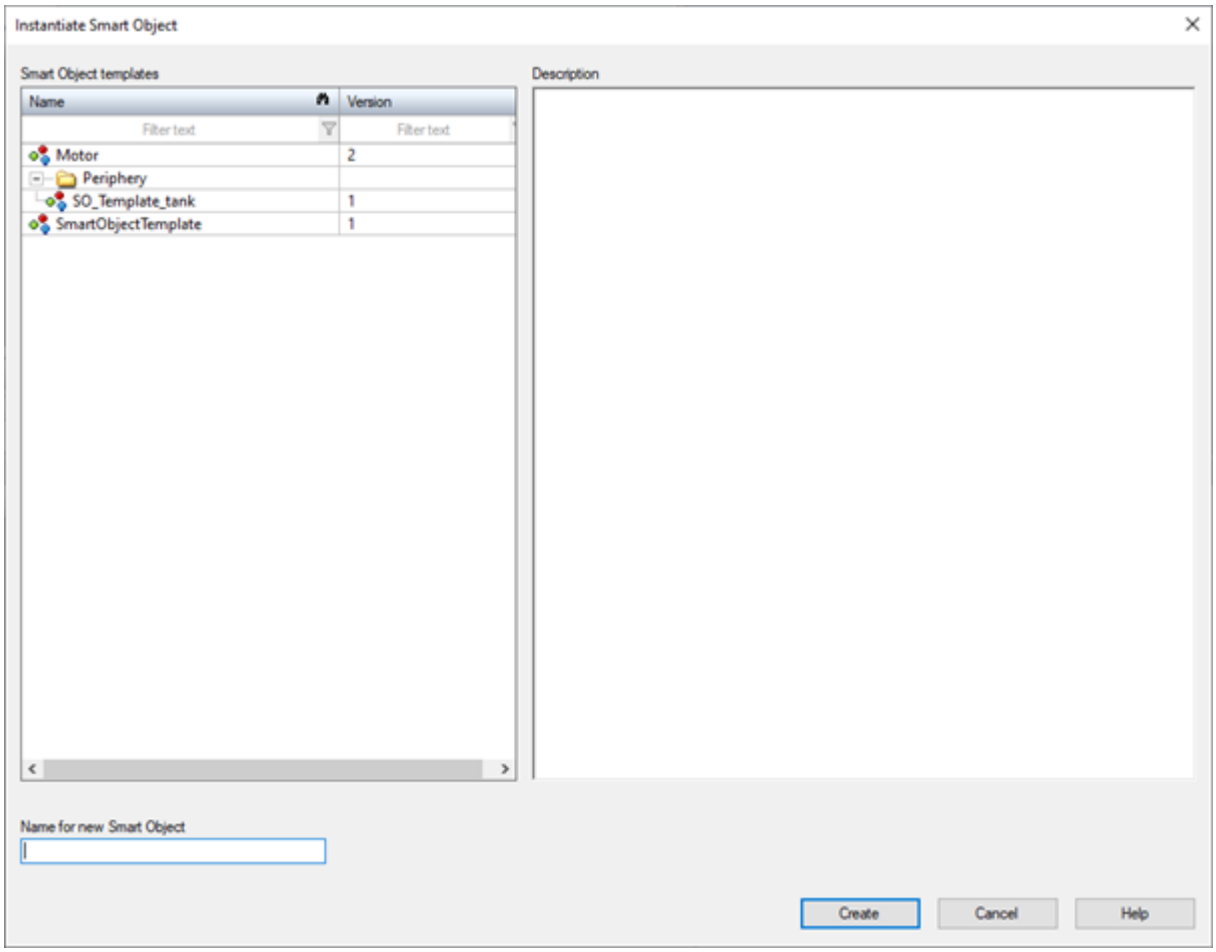
The following options are available in the toolbar and the context menu for the configuration of smart objects:

Parameters	Description
Create new instance	Creates a new smart object based on existing smart object templates. For this, the **Instantiate smart object** (on page 9) dialog will be opened. **Note:** If the smart object template contains linked variables, the dialog for mapping variables (on page 13) will then open afterwards.
Manage smart object templates	Manages smart object templates for use in zenon configurations. For this, the **Manage smart object templates** (on page 11) dialog will be opened.
Copy	Copies the selected entries to the clipboard.

Parameters	Description
Paste	Pastes the content from the clipboard. The pasted smart object template is given the name of the copied original object plus an additional number.
Delete	Deletes the selected smart object. For this, the dialog to delete (on page 15) the selected smart object will be opened. Multiple selection is possible.
Expand all	Displays the entire tree structure. By clicking on the arrow you receive a drop-down list in which you can select one of the following commands: ▸ Expand all: expands all nodes ▸ Collapse all: collapses all nodes ▸ Expand selection: expands selected nodes ▸ Collapse selection: collapses selected nodes A click on the button always expands all elements. **Hint:** Via double click on the superordinate entry elements can also be expanded or collapsed.
Enter the properties	Enters the properties for the selected element. These inputs are parameterized in the properties window. If the properties window is hidden, it will be opened in the Editor.
Help	Opens online help.

4.2 “Instantiate smart object” dialog

Smart objects are created in this dialog. A smart object is always based on a smart object template. After selecting this template, the smart object to be created must be named properly.



SMART OBJECT TEMPLATES

List of the available smart object templates. This list contains all the smart object templates contained in the project.

The context menu can be used to hide or display columns. You can access this context menu by right clicking in a column header.

Status information about available and selected elements is shown as text in the footer of the list.

Parameter	Description
Name	Current name of the smart object template.
Version	Current version of the smart object template.

DESCRIPTION

This description is contained in the smart object template. The author of the template is responsible for creating a description for the smart object template. If the template does not contain such a description, this area of the screen will remain empty.

NAME FOR NEW SMART OBJECT

In this input field, you enter the name of the smart object in the zenon configuration.

The entry is validated. If the input is invalid, a corresponding message will be shown.

The name must:

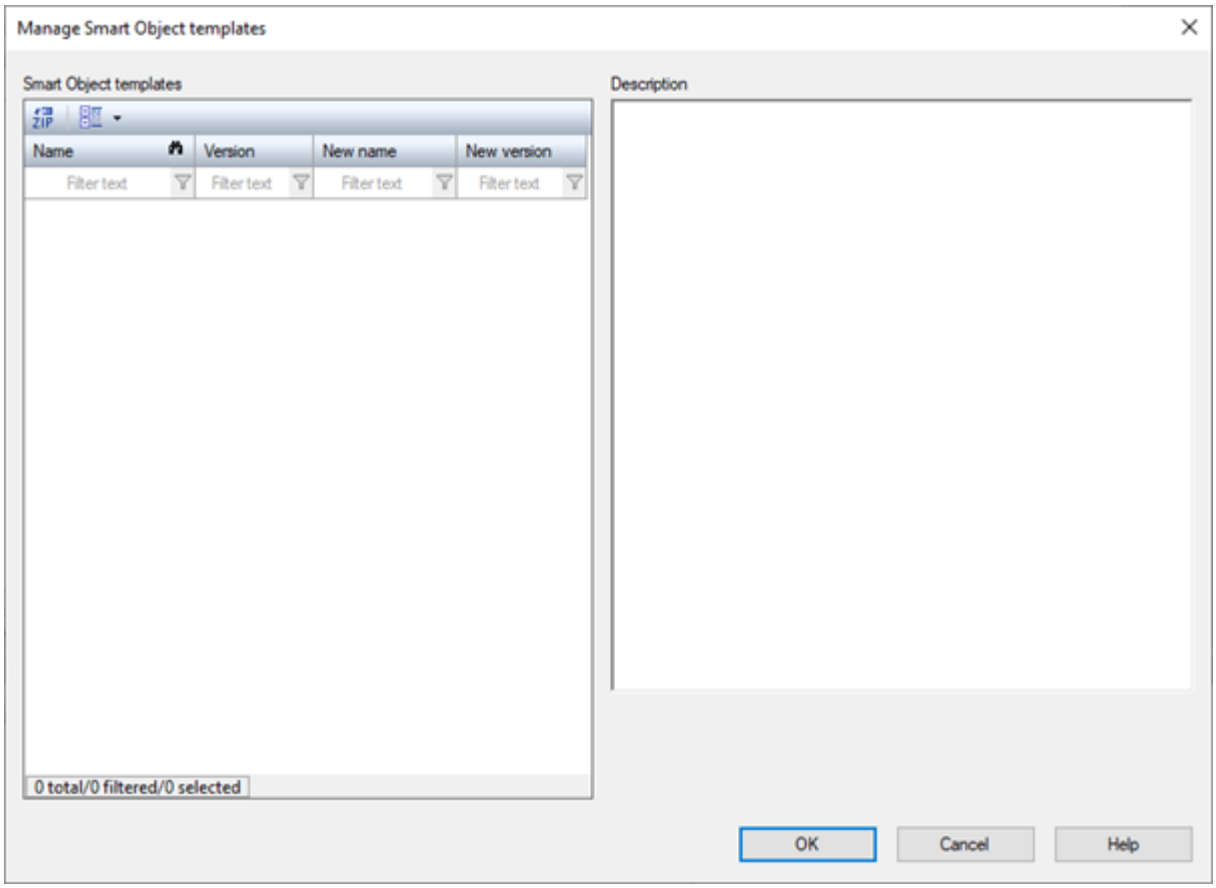
- Not be used already for another smart object
- Not be empty
- Not begin with a number
- Not end with an underscore
- Not contain two consecutive underscores
- Not contain any spaces
- Not contain any special characters: @""!§$%&/()=?*';:-<>
- Not contain any language-dependent characters

CLOSE DIALOG

Parameter	Description
Create	Creates a smart object and closes the dialog. To create the object, a smart object template must be selected from the list and a correct name for the smart object to be created must be configured in the input field. The current parameterization is validated. A warning message is shown in the dialog if an error occurs.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.

4.3 “Manage smart object templates” dialog

In this dialog, smart object templates are managed and made available for configuration in the project tree of the zenon Editor.



SMART OBJECT TEMPLATES

Parameter	Description
Import	Opens the file selection dialog to select a saved .so file
Expand/ collapse node	Opens the drop-down list list to expand or collapse nodes.
▸ **Expand all**	Expands all nodes.
▸ **Collapse all**	Collapses all nodes.
▸ **Expand selection**	Expands the highlighted nodes.

Parameter	Description
▸ **Collapse selection**	Collapses the highlighted nodes.

LIST OF THE SMART OBJECT TEMPLATES LOADED IN THE PROJECT

Parameter	Description
Name	Current name of the smart object template.
Version	Current version of the smart object template. Default: *0**
New name	New name of the smart object template after import.
New version	New version number of the smart object template after import.

> **Attention**
>
> Previously imported smart object templates can now be updated.

DESCRIPTION

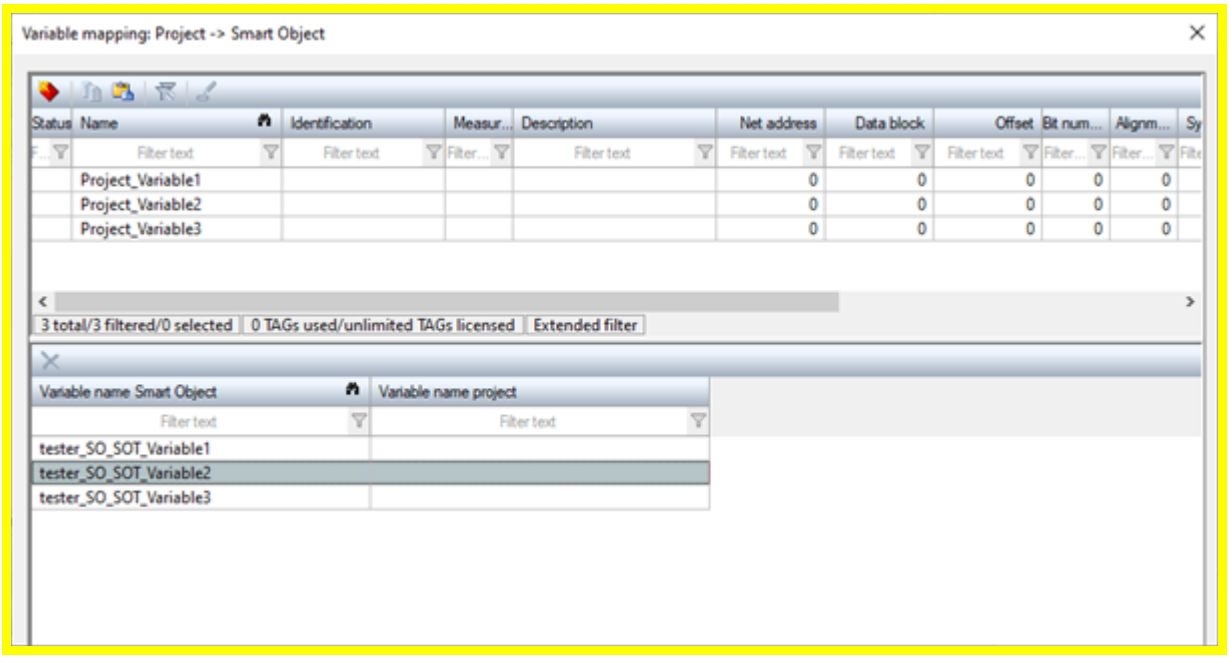
This description is contained in the smart object template. The author of the template is responsible for creating a description for the smart object template. If the template does not contain such a description, this area of the screen will remain empty.

CLOSE DIALOG

Options	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.

4.4 “Map variables” dialog

In this dialog, variables existing in the zenon project are mapped to the variables configured in the smart object template. The variables mapped in the **Project variable name** column are used in productive operation and for display in the zenon Runtime. The variables are mapped via drag & drop.



The dialog is divided into two areas:

- Variables of the current zenon project
- Variables configured in the smart object template in use and current project variables

TOOLBAR OF THE DIALOG

Parameter	Description
New variable	Opens the dialog to create a new variable for the current zenon project.
Copy	Copies the selected entries to the clipboard.
Paste	Pastes the content from the clipboard. If an entry with the same name already exists, the content is pasted as "**Copy of...**".
Remove all filters	Removes all filter settings. **Note:** Only active if the current view is filtered.
Edit selected cell	Opens the selected cell for editing. The binocular

Parameter	Description
	symbol in the header shows which cell has been selected in a highlighted line. Only cells that can be edited can be selected.

LIST - PROJECT VARIABLES

List of the variables already existing in the current zenon project

The context menu can be used to hide or display columns. You can access this context menu by right clicking in a column header.

Status information about available and selected elements is shown as text in the footer of the list.

MAP VARIABLES

In this area, variables of the current zenon project are mapped to the variables configured in the smart object template. The variables mapped in the Project variable name column are used in productive operation and for display in the zenon Runtime.

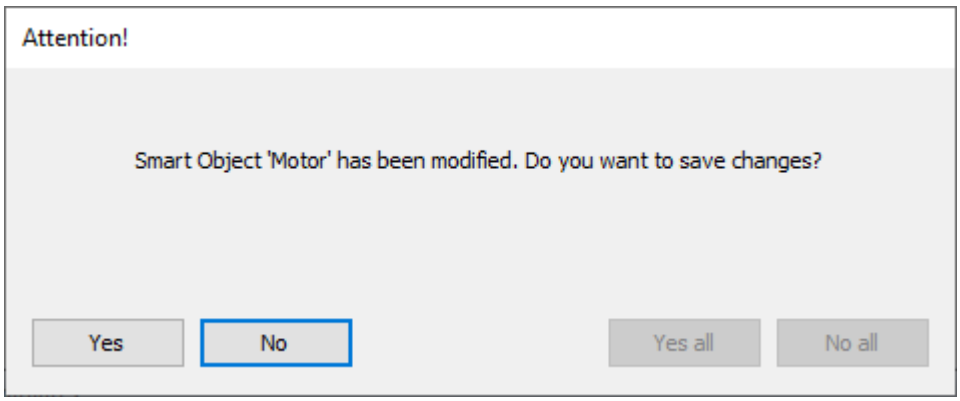
Note: If the **Variables interchangeable** property has been deactivated during the configuration of a variable in the smart object template, this variable will not be shown in the list. It cannot be mapped to a project variable.

Parameter	Description
Clear mapping (red X symbol)	Deletes the mapping of the selected project variable. Multiple selection is possible.
Smart object variable name	Overview of the variables configured in the smart object template. These entries cannot be changed.
Project variable name	Mapped variable of the current zenon project. Variables are mapped via drag & drop The mapping can be cleared using the **Delete** symbol.

CLOSE DIALOG

Options	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.

4.5 “Delete smart object” dialog



Parameter	Description
Yes	The currently displayed smart object or smart object template is saved. If several smart objects or smart object templates contain unsaved changes, these are also displayed in this dialog after confirmation. After confirmation of the last smart object or smart object template with unsaved changes, the dialog closes.
No	The currently displayed smart object or smart object template is not saved. If several smart objects or smart object templates contain unsaved changes, these are displayed in this dialog after clicking the button. After clicking on the button of the last smart object or smart object template with unsaved changes, the dialog closes.
Yes all	All smart objects or smart object templates with unsaved changes are saved. The dialog closes afterwards. **Note:** Only available if several objects contain unsaved changes.
No all	All smart objects or smart object templates with unsaved changes are not saved. The dialog closes afterwards. **Note:** Only available if several objects contain unsaved changes.

5 Manage and parameterize smart object templates

Templates for smart objects are configured in the **Smart Object templates** tab.

Smart object templates can be exported from and imported into any project. This allows configuration contents to be exchanged between several computers or project engineers.

Information

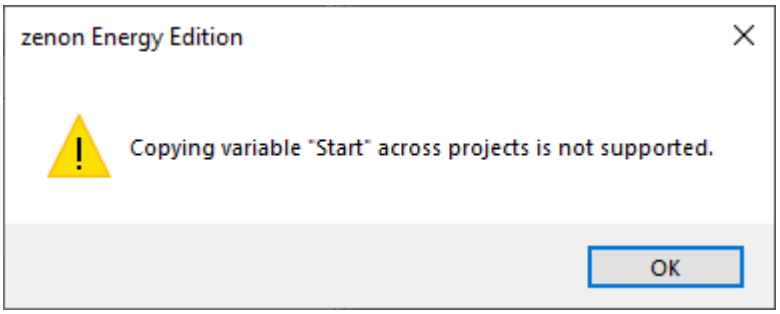
You can find information on the configuration and parameterization of smart objects in the Configure and parametrize smart objects (on page 6) chapter.

RESTRICTIONS

The following restrictions apply for the configuration of smart object templates:

- The Smart Object module does not support **distributed engineering**. Smart Object contents created locally are not transferred to the project server. If a multi-user project is synchronized, the local smart object contents are deleted.
- Copy and paste/drag & drop
  - It is not possible to copy and paste configuration contents between a zenon project and a smart object configuration. This prevents changes from leading to misconfigurations such as incorrect links.
  - Configuration contents can only be copied and pasted/dragged and dropped within a smart object template. A configured zenon element cannot be exchanged between two smart object templates.

  Note: This restriction is also visualized by a dialog in the zenon Editor.

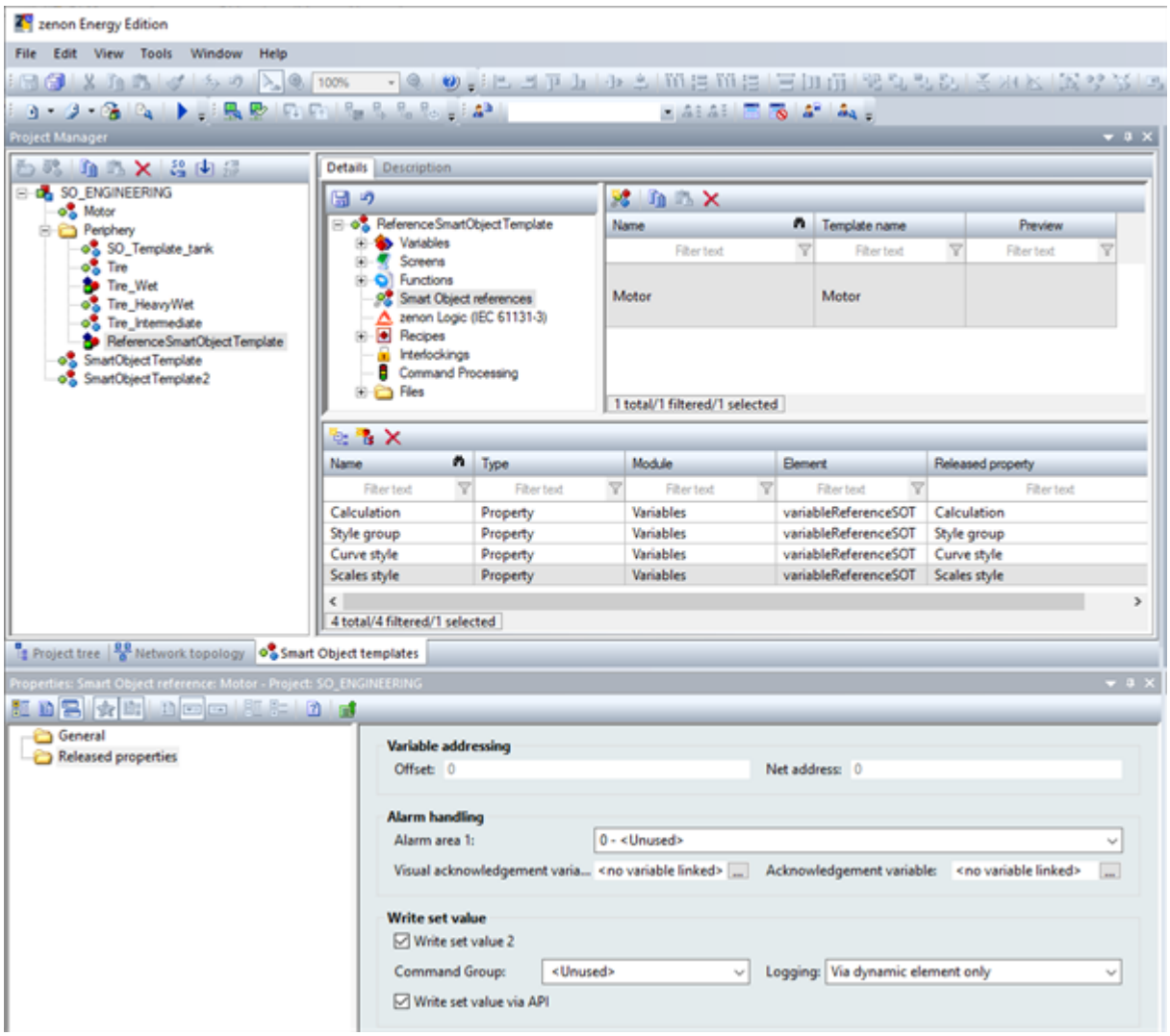


5.1 Manage smart object templates

The smart object templates are managed and configured in a separate tab of the project manager.

This view contains the engineering environment for smart object templates. The templates are created and managed in this view separately from the configuration in the zenon Editor. The

appearance is based on the project tree but only contains those modules and functionalities necessary for creating a template for a smart object.



The view of the engineering environment of the smart object template is divided into the following parts:

- Overview of smart object templates (on page 19)

  All the existing smart object templates are listed and managed in a tree view.

- Main window (on page 24)
  - Engineering environment of smart object templates (on page 25)

    In a second tree view, all the modules are shown that support configurations for a smart object template.

  - Released properties (on page 30)

    The zenon properties are managed in this view. Only properties that are configured here are available to the user of a smart object for parameterization.

- Description (on page 33)

  Additional user-defined information for the smart object template.

5.1.1 Versioning

Every smart object template contains a version number. This number is given by zenon automatically and cannot be changed. The current version number is displayed in the **Version** property of the smart object template.

The following applies to versioning:

- Version numbers for new smart object templates always begin with *0**
- Each time the template is exported, the version number is increased by the value *1*.
- If no changes have been made to the configuration since the last export, the version number is not increased.
- If the configuration contains changes that have not yet been exported, this is indicated by a * character.
  Example: *1** - changes have been made to the smart object template since the last export. However, it has not been exported since then.

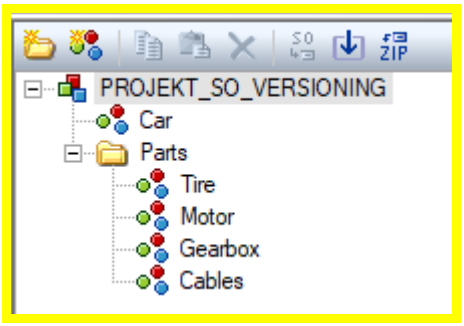
VERSIONING OF STORAGE FILES

If a smart object template is saved via export, the storage file always contains the most recent version of the smart object template. Versioning within the .so storage files is not planned. Therefore, you should save each version of the template with a unique name. The mapping to the configured smart objects is always done using the unique **GUID** of the smart object template.

> **Hint**
>
> When exporting a smart object template, you should always give a unique name that also contains a version information such as *motor_V3.so*.

5.1.2 Tree view

All the existing smart object templates are listed and managed in a tree view.



In this view, smart objects templates are:

- Created and deleted
- Stored in a tree view in a structured manner
- Imported from or exported to a file
  Note: Smart object templates are saved in their own .SO file format. Import and export via XML is not possible.
- Arranged via drag & drop
- Copied and pasted

> **Information**
>
> You can find further information on this in the Manually create a smart object template (on page 45) chapter in this manual.

5.1.2.1 Context menu and toolbar

The following options are available in the toolbar and the context menu for the configuration of smart object templates:

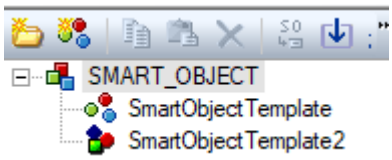
Parameter	Description
Create folder	Creates a new folder in the tree view. The new folder is added at the end of the selected node. If an existing folder is selected during the creation of a folder, the new folder is created as a subfolder.

Parameter	Description
	The new folder is named *Folder* by default. If necessary, a consecutive number is added to this default name. **Note:** This is only active if the first level of the node or an existing folder is selected.
Create smart object	Creates a new smart object template in the currently selected storage level. The new smart object template is named *SmartObjectTemplate* by default. If necessary, a consecutive number is added to this default name.
Copy	Copies the selected entries to the clipboard. **Note:** This is only active if a smart object template is selected. Smart object templates can only be copied and pasted within a zenon project.
Paste	Pastes the content from the clipboard. If an entry with the same name already exists, the content is pasted as "**Copy of...**". The content is added to the selected level as a new smart object template with its own GUID. If the copied smart object template contains released properties, these are also contained in the copy. **Note:** This is only available if a smart object template has been copied to the clipboard beforehand. Smart object templates can only be copied and pasted within a zenon project.
Delete	Deletes selected entries after a confirmation from list. If a folder of the tree view is deleted, all the smart object templates contained within it will also be deleted. **Note:** This is only available if a smart object template or a folder has been selected in the tree view beforehand.

Parameter	Description
	Multiple selection is not possible.
Export	Opens the dialog to save a file and exports the selected smart object template. **Note:** This is only available if a smart object template is selected.
Export all	Exports all the smart object templates contained in the tree. Every configured smart object template is saved as its own .SO file. It is not possible to limit this to the smart object templates you want to export. It always exports all the smart object templates contained in the tree. If, for example, the folder in a subordinate hierarchy is selected in the tree view, the export will contain all the smart object templates. The saving location of the export can be selected via a dialog. **Attention:** Pre-existing .SO files are overwritten without prompting. If the tree contains templates with the same name, only one file will be saved! This file contains only the configuration of the latest smart object template with the same name in the tree.
Import	Opens the **Manage smart object templates** (on page 11) dialog. Selection dialog for importing a smart object storage file. After the file is selected, the imported smart object template is added to the selected level. **Note:** This is only active if a folder or the first level of the node is selected in the tree view.

5.1.2.2 Unsaved changes

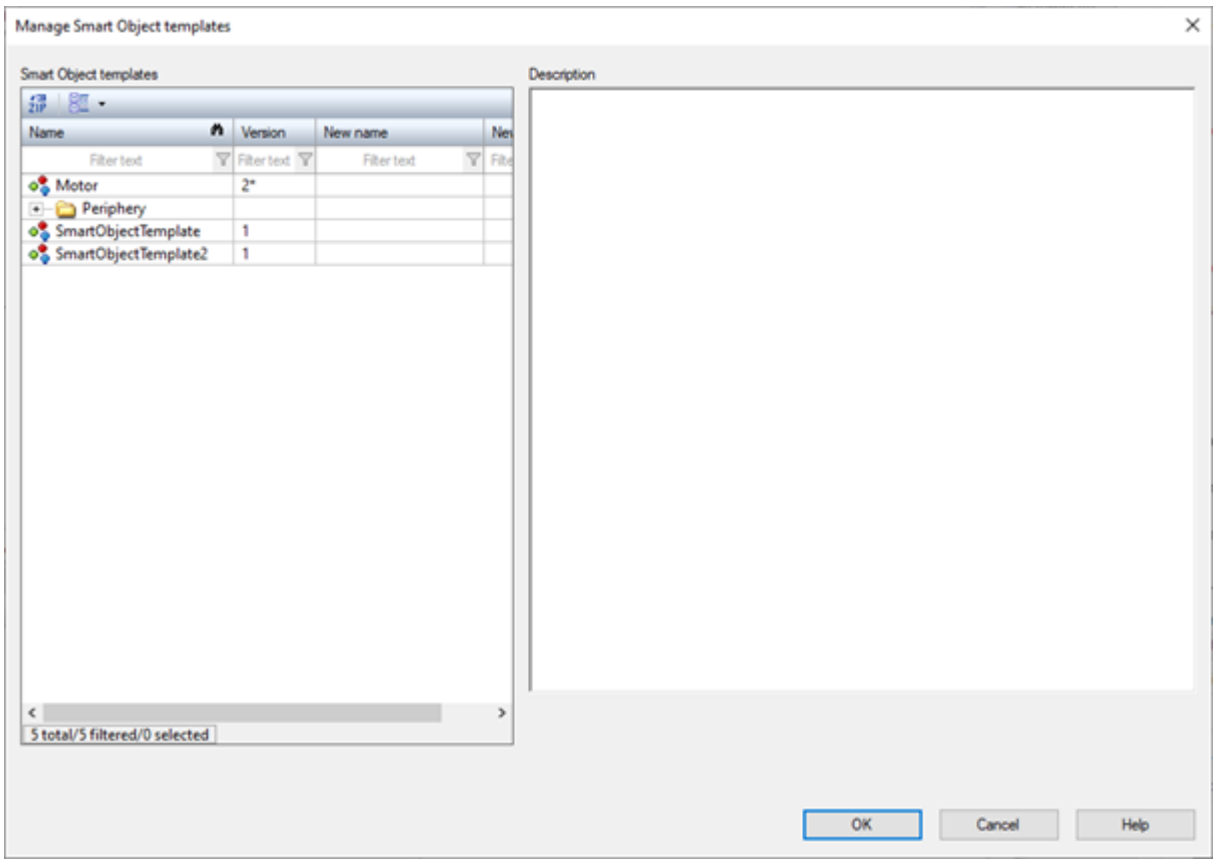
Smart object templates with unsaved changes are indicated by a symbol:



Symbol	Description
	The smart object template does not contain any unsaved changes.
	Changes have been made to the configuration or parameterization of the smart object template since it was last saved.

5.1.2.3 Import smart object templates

In this dialog, smart object templates are imported.



SYMBOLS - SMART OBJECT TEMPLATES

Parameter	Description
Import	Opens the file selection dialog to select a saved .SO file.
Expand/ collapse node	Opens the drop-down list list to expand or collapse nodes.
▸ **Expand all**	Expands all nodes.
▸ **Collapse all**	Collapses all nodes.
▸ **Expand selection**	Expands the highlighted nodes.
▸ **Collapse selection**	Collapses the highlighted nodes.

LIST - SMART OBJECT TEMPLATES

List of configured smart object templates. This list can be sorted and filtered.

The context menu can be used to hide or display columns. You can access this context menu by right clicking in a column header.

Status information about available and selected elements is shown as text in the footer of the list.

Parameter	Description
Name	Current name of the smart object template.
Version	Current version of the smart object template. Default: *0**
New name	New name of the smart object template after import.
New version	New version number of the smart object template after import.

DESCRIPTION

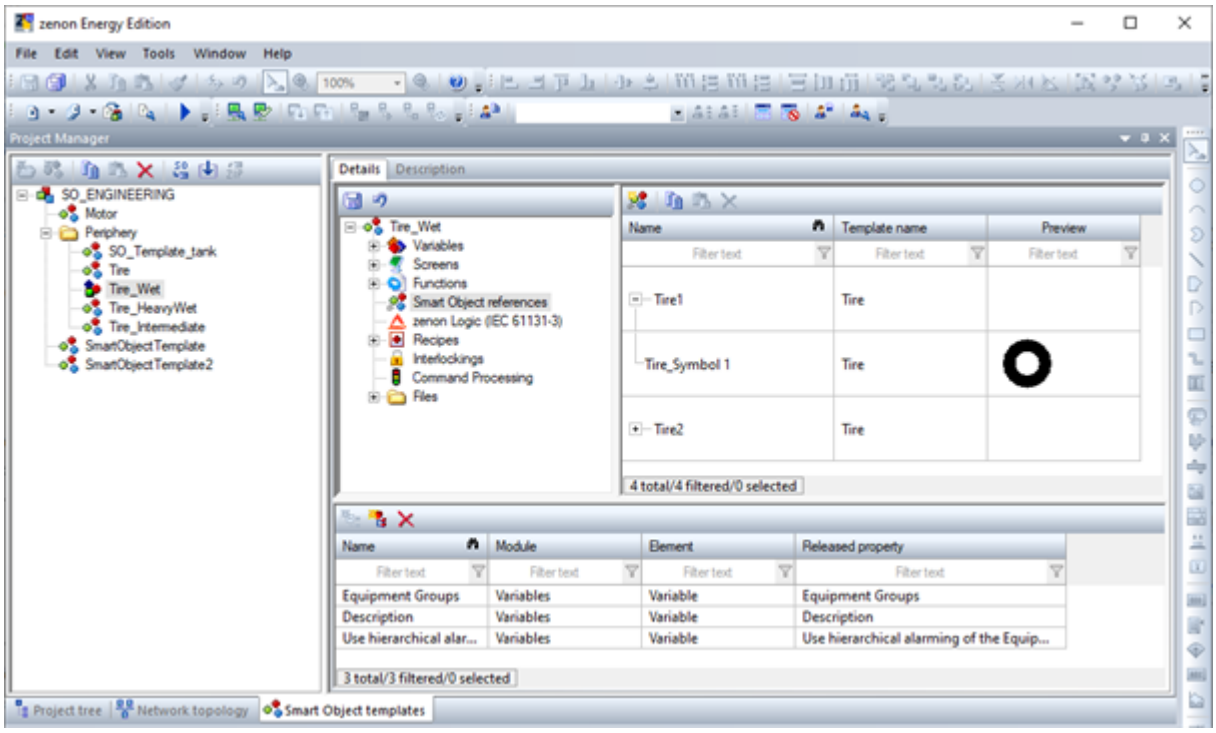
In this area, information on the selected element is displayed. This description is contained in the smart object template. The author of the template is responsible for creating a description for the smart object template. If the template does not contain such a description, this area of the screen will remain empty.

CLOSE DIALOG

Options	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.
Help	Opens online help.

5.1.3 Main window

Smart object templates are configured in the main window.



- Details (on page 25)
  Engineering environment for smart object templates.
- Description (on page 33)
  Preview window with a description of the selected smart object template.
  This description is contained in the smart object template. The author of the template is responsible for creating a description for the smart object template. If the template does not contain such a description, this area of the screen will remain empty.
  You can find further information on this in the Description chapter.
- Released properties (on page 30)

List of the released properties that are available to the user for the parameterization of a smart object.

5.1.3.1 Details - Tree view for smart object templates

In the tree view, all the modules are shown that support configurations for a smart object template. The toolbars and menu entries are the same as those in the project tree of the Editor.

TOOLBAR

Parameter	Description
Save changes	Saves the currently selected smart object template and applies all the new configurations and changes.
Discard Changes	Restores the smart object template to the state it was in before the last saving. In the case of new smart object templates, their content is deleted in the process because they have not been saved yet.

Information

If the active project is modified or the <CD PRODUCTNAME> Editor is closed and the smart object templates contain unsaved changes, a dialog with the option to save these changes will be displayed.

5.1.3.1.1 Variables

Smart objects support the following functionality of the node's variables:

- Driver
- Data Types
- Reaction matrix

Information

You can find further information in the Variables manual and in the linked chapters of this manual.

5.1.3.1.2 Screens

In this node, you can configure the screen types and the symbol library for smart object templates.

This symbol library contains only the symbols of the selected smart object template. There is no direct link to the symbol library in the project tree or to other smart object templates. If additional smart object templates are referenced in the current smart object template, the screens used within them are also displayed in the current template.

SUPPORTED SCREEN TYPES

The following screen types are supported for a smart object template:

- **Faceplate**
  This screen type offers a frame with four screen containers.
  You can find further information on this in the Faceplate chapter in the Screens manual.
- **Standard**
- **Alarm Message List**
- **Extended Trend**
- **Command Processing**
- **Chronological Event List**

> **Information**
>
> You can find general information in the Screens manual and in the linked chapters of this manual.

5.1.3.1.3 Functions

Smart Objects support all the functions that are also available for configuration in the zenon project tree.

> **Information**
>
> You can find general information in the Functions manual and in the linked topics of the Overview of functions in zenon chapter.

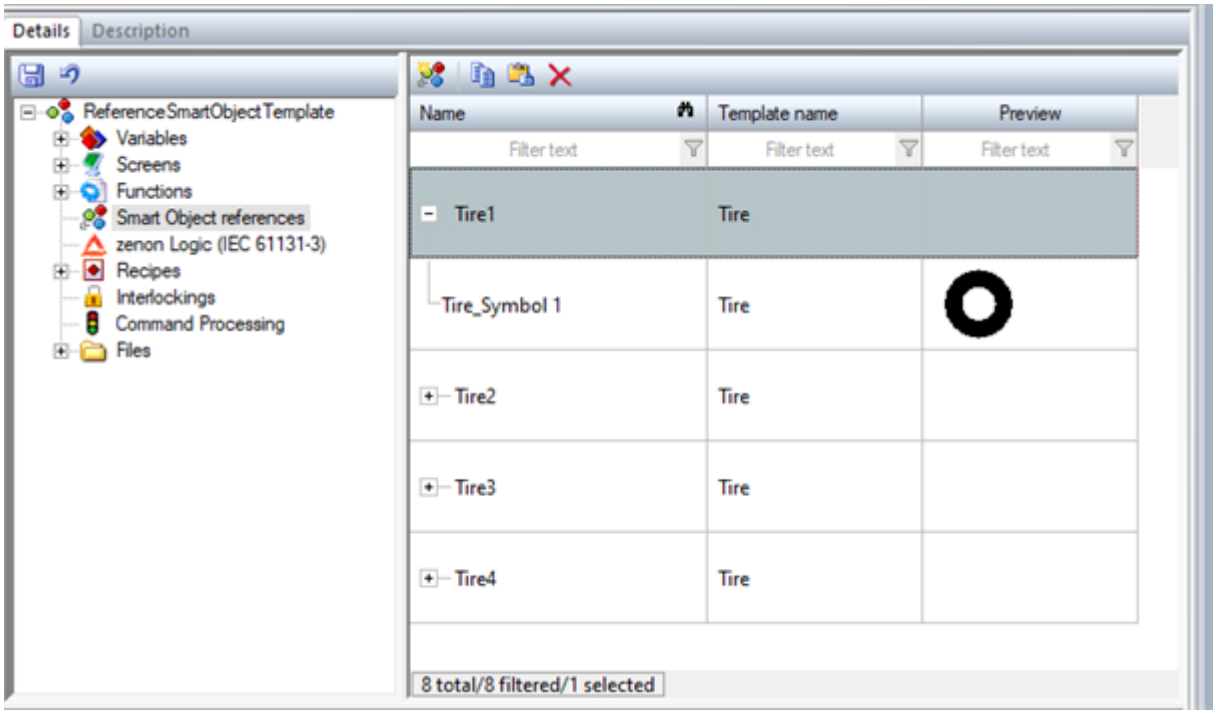
SCRIPTS

In addition to functions, scripts are also available for the configuration of a smart object template.

> **Information**
>
> You can find information on scripts in the Scripts chapter in the Functions manual.

5.1.3.1.4 Smart Object references

In this node, you can link a smart object template to the configuration contents of one or several other smart object templates. This allows you to carry out nested, non-recursive configurations. Smart object templates apply all the content of those smart object templates for which a reference has been configured.



Referenced smart object templates provide maximum flexibility in the development of smart object templates without thereby increasing the complexity for end users. In this way, smart object templates can refer to each other on several hierarchy levels. A smart object template can thus be part of another smart object template. Templates can be built in modules, divided into smaller applications and be reused. The following applies here: The smart object template applies all the parameterizations of the smart object template for which a reference has been configured. Released properties of the linked smart object template must be released again for the current smart object template.

A smart object template can contain several links to the same or several different smart object templates.

Example

A smart object template represents a *car*. Another smart object template represents a *tire*. A link to the *tires* smart object template is configured in the *car* smart object template for each tire. If the configuration of the *tires* smart object template is changed, these changes are applied automatically in the *car* smart object template.

Advantages:

- Simplified maintenance of a smart object template with large amounts of configuration content.
- Reduction of complexity by dividing the templates into smaller sub-areas
- Better overview by dividing the templates into segments (e.g., pumps, tanks valves, or cars and tires)
- Division of work among several project engineers.
- Reusability for different applications since the entire configuration content is not collected in one object.

Information

You can find further information and step-by-step instructions in the **Reference smart object templates** (on page 57) chapter in this manual.

5.1.3.1.5 Context menu and toolbar

The following options are available in the toolbar and the context menu for the referencing of smart object templates:

Parameter	Description
Reference smart object	References a smart object template with the content of another smart object template. Clicking on the symbol or selecting the option in the context menu opens the dialog for creating a new Smart Object reference. **Note:** A smart object template can contain several

Parameter	Description
	references.
Copy selected elements to the clipboard	Copies the selected entries to the clipboard.
Insert contents from clipboard	Pastes the content from the clipboard. If an entry with the same name already exists, the content is pasted as "**Copy of...**".
Delete reference	Deletes the reference to the base smart object template. The smart object template referred to, however, is not deleted and remains unchanged.

CONTEXT MENU

The context menu provides the following options for configuring references:

Parameter	Description
New reference...	Opens the **Instantiate smart object** (on page 9) dialog for referencing an existing smart object template.
Help	Opens online help.

5.1.3.1.6 zenon Logic

This node provides the zenon functionality of zenon Logic for smart object template configurations.

> **Information**
>
> You can find further information in the <CA_PRODUCTNAME >manual.

5.1.3.1.7 Interlockings

Smart object templates support the full scope of the **Interlockings** module as they are available for configuration in the zenon project tree.

> **Information**
>
> You can find general information on this in the Interlockings manual.

5.1.3.1.8 Command Processing

Smart object templates support the full scope of the **Command processing** module as they are available for configuration in the zenon project tree.

> **Info**
>
> You can find general information on this in the Command processing manual.

5.1.3.1.9 Files

The following external files can be integrated into a smart object template via the **Files** node of the tree view:

- Graphics
- Help
- Multimedia
- Drivers
- Miscellaneous

The node and the supported subnodes only contain files in the smart object templates tree view. There is no direct link to the files in the project tree.

Not supported:

- Texts and Formats
- Report Viewer

> **Information**
>
> You can find general information on this in the Editor manual and in the linked themes of the Files chapter.

5.1.3.2 Released properties

All properties that are configured in the smart object template are basically blocked for the user of the smart object in the zenon project.

Releases can be created for properties of the configuration contents of a smart object template so the user of the template has some limited configuration options. These released properties can then be parameterized by the user of a smart object.

A smart object always contains parameterizations of released properties for one or several zenon elements. Project properties are not parameterized by smart objects. These are configured in the zenon project properties, independently of the **Smart Objects** module.

Information

You can find further information and step-by-step instructions in the Edit smart object templates (on page 48) chapter in this manual.

5.1.3.2.1 Context menu and toolbars

The following options are available in the tool for the configuration of released properties of smart object templates:

Parameter	Description
New group	Merges entries into one group. Clicking on the symbol merges all the highlighted elements in the list into one group, provided they are compatible. For this, a new node is created in which all the highlighted entries are listed as subentries.
New node	Creates a new node.
Delete	Deletes the highlighted entry in the list. Multiple selection is possible. The highlighted entries are not deleted immediately. A confirmation dialog is opened before deletion.

LIST OF RELEASED PROPERTIES

List of released properties for the smart object template.

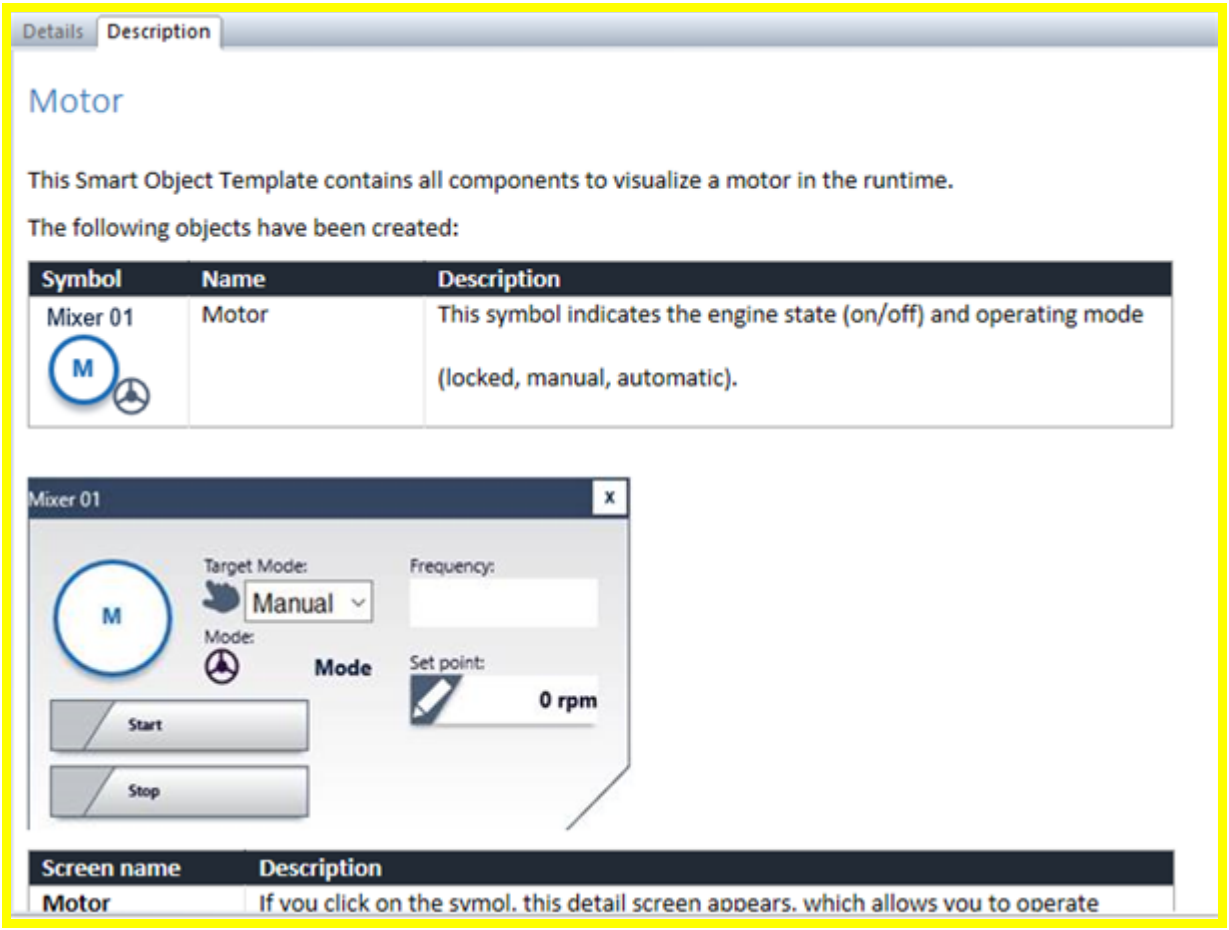
The context menu can be used to hide or display columns. You can access this context menu by right clicking in a column header.

Status information about available and selected elements is shown as text in the footer of the list.

Parameter	Description
Name	Name of the released property.
Module	Origin of the parameterization. This corresponds to the name of the node in the tree view as this is displayed in the **Details** tab of the smart object template. This information is assigned automatically and cannot be changed.
Element	Configured element that uses this property. This information is assigned automatically and cannot be changed. **Hint:** To find the element, go to the node in the Details tab as shown in the Module column.
Released property	Original name of the released property. This information is assigned automatically and cannot be changed.
Type	Type of entry in the list of released properties. ▸ *Group*: Merging of several properties with the same configuration content into one property. Only one property with the name of the group is displayed in the smart object. This parameterization is then used for all the properties linked with this group. ▸ *Property*: List element corresponds to a zenon property. ▸ *Node*: Structuring of properties as a group box. The node name is displayed as the name of the group box when configuring a smart object. **Note:** This column is hidden by default. This information is assigned automatically and cannot be changed.

5.1.4 Description

You can find a short overview of the configured smart object template in the **Description** tab.



When a smart object template is created for the first time, it does not have a description. This description must be created outside of the zenon environment in .HTML format.

Information

You can find further information and step-by-step instructions in the **Create a description for a smart object template** (on page 59) chapter in this manual.

FILE FORMAT

The start page for the description must be named *index.htm*. Internet-compatible files can be used for the description. You can also use, for instance, .CSS files and a file structure with subfolders (in the respective language folder).

SAVING LOCATION

All the configuration content of a smart object template is saved in the following saving location:

C:\ProgramData\COPA-DATA\SQL2016\[Project GUID]\FILES\zenon\system\SmartObjects\[Smart Object Template GUID].

When a smart object template is created, a *Description* subfolder is automatically created. The respective subfolders for the languages available in zenon are automatically created in this subfolder. The description in the respective language is saved in this language folder.

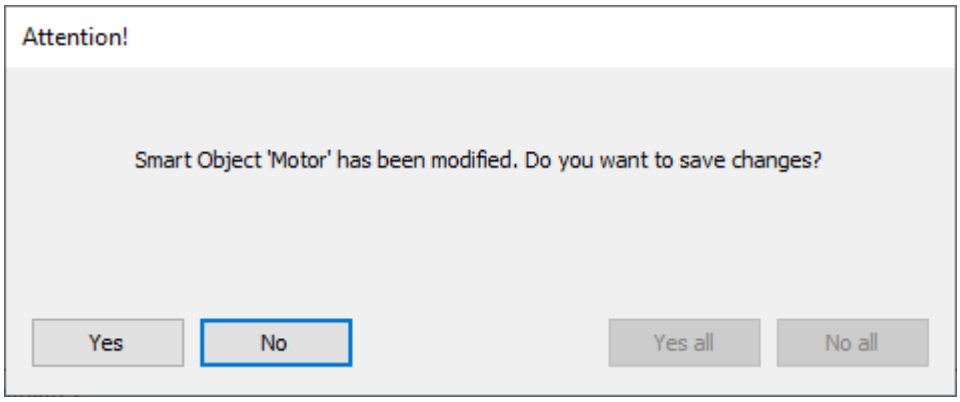
If the zenon Editor is started in a language and the smart object template description is empty in the respective language folder, the content of the *ENGLISH* folder is shown by default. If this folder is also empty, then the description is displayed as empty.

Example

If the Editor is opened in Japanese, for example, and the respective language folder of the smart object template (*JAPANESE*) does not contain any content, the content of the *ENGLISH* folder will be displayed in the Editor.

5.1.5 "Delete smart object template" dialog

This dialog is displayed if you delete one or several smart objects or smart object templates.



Parameter	Description
Yes	The currently displayed smart object or smart object template is saved. If several smart objects or smart object templates contain unsaved changes, these are also displayed in this dialog after confirmation. After confirmation of the last smart object or smart object template with unsaved changes, the dialog closes.

Parameter	Description
No	The currently displayed smart object or smart object template is not saved. If several smart objects or smart object templates contain unsaved changes, these are displayed in this dialog after clicking the button. After clicking on the button of the last smart object or smart object template with unsaved changes, the dialog closes.
Yes all	All smart objects or smart object templates with unsaved changes are saved. The dialog closes afterwards. **Note:** Only available if several objects contain unsaved changes.
No all	All smart objects or smart object templates with unsaved changes are not saved. The dialog closes afterwards. **Note:** Only available if several objects contain unsaved changes.

6 Engineering in the Editor

In this section of the manual, you can find practical engineering tips for your work in the zenon Editor.

- Use smart object (on page 36)
  In this section, you can find detailed engineering instructions on how you can use and customize existing smart object templates in your zenon.
- Configuration of smart object templates (on page 45)
  In this section, you can find detailed instructions on how you can merge configuration contents into one smart object template.

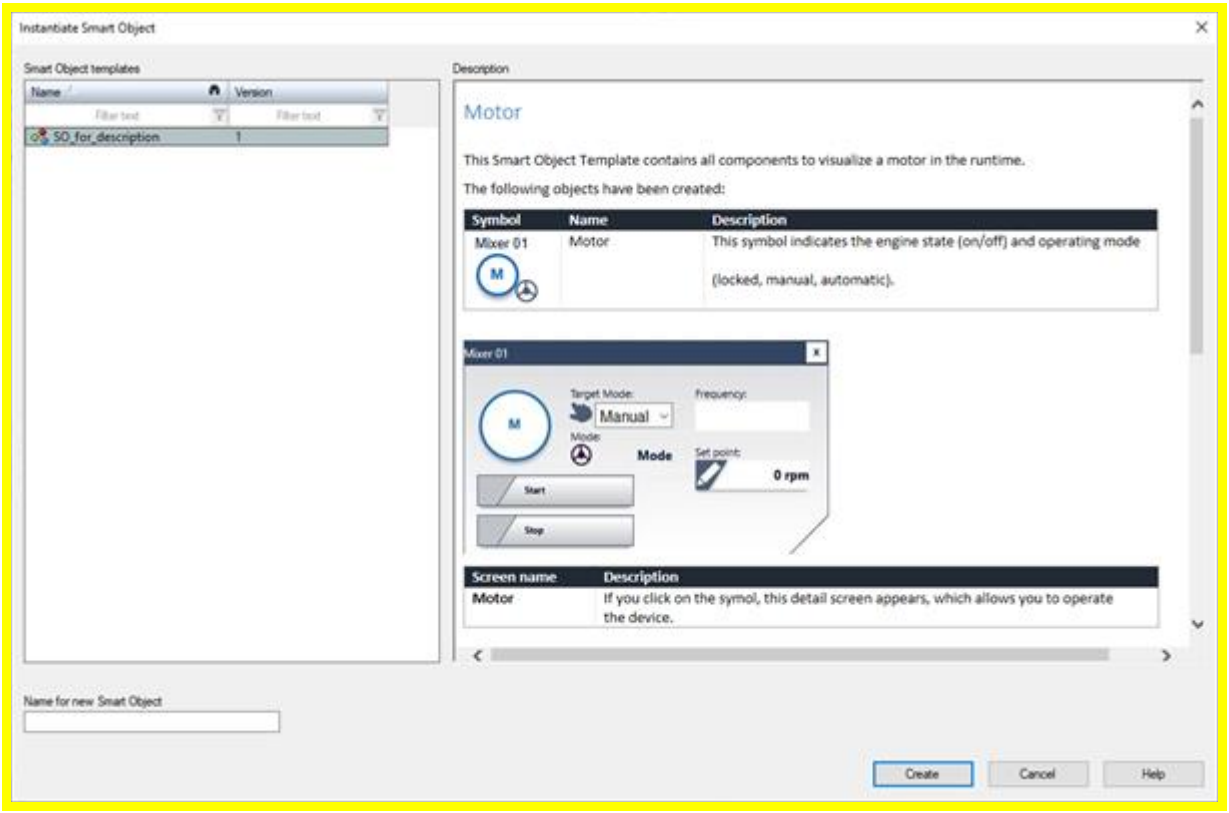
> **Attention**
>
> The Smart Object module does not support **distributed engineering**. Smart Object contents created locally are not transferred to the project server. If a multi-user project is synchronized, the local smart object contents are deleted.

6.1 Smart objects

When a smart object is created, the configuration contents that have been merged and pre-configured in a smart object template are applied in a zenon project. In this section, you can find engineering instructions as well as tips on the creation, management and parameterization of smart objects.

6.1.1 Create smart object

Follow the following steps to create a smart object:



1. Switch to the **Project tree** tab in the project manager.
2. In the project tree, select the **Smart Objects** node.
3. Create a new smart object.
   a) To do this, click on the symbol or the **Create new instance** entry in the context menu.

      This opens the **Instantiate smart object** (on page 9) for selecting a smart object template.
   b) Select an existing smart object template from the **Smart object templates** list as the basis for the smart object.
   c) Give the new smart object to be created a name by entering a name in the **Name for new smart object** input field.

Note: The name of the smart object can be changed at anytime via the **Name** property.

d) Confirm your configuration by clicking on the **Create** button.

4. The **Variable mapping: Project -> Smart object** dialog is opened.
   - At the top of the dialog, all the configured variables of the current zenon project are listed.
   - At the bottom of the dialog, all the variables are listed which are available via the smart object.
5. Map the project variables to the configured variables of the smart object via drag & drop.

   Note: If the smart object does not contain any variables, you can skip the mapping of variables in the dialog by clicking the **OK** button.

   The following applies for the mapping of variables:

   - A project variable can only ever be mapped to one smart object variable. Multiple selection is not possible.
   - Previously existing configured mappings can be cleared by clicking the symbol or the **Clear mapping** entry in the context menu.
   - A previously configured mapping is replaced by dragging & dropping another project variable.
   - Variables can be remapped at anytime using the **Variable mapping** smart object property.

   Attention: Please note that only simple data types are available for mapping. While structured data types are available for the configuration of smart objects, the address information of the variables based on these cannot be replaced by the variable mapping functionality. Configured structured data types are not displayed in the list of project variables.
6. Confirm this mapping by clicking the **OK** button.
7. The smart object is displayed in the detail list of the **Smart Object** node.
8. Click to select the smart object and parameterize the **Released properties**.
9. Place the smart object in our current zenon configuration.

Information

Instead of mapping the variables using the dialog, you can also carry out an individual addressing of the variables of a smart object by performing the following engineering steps. The properties of the **Addressing** variable properties group (for example, **Net address**, **Offset**, **Symbolic address**, ...) are configured as released properties in the smart object template on which the smart object is based.

6.1.2 Parameterize smart object

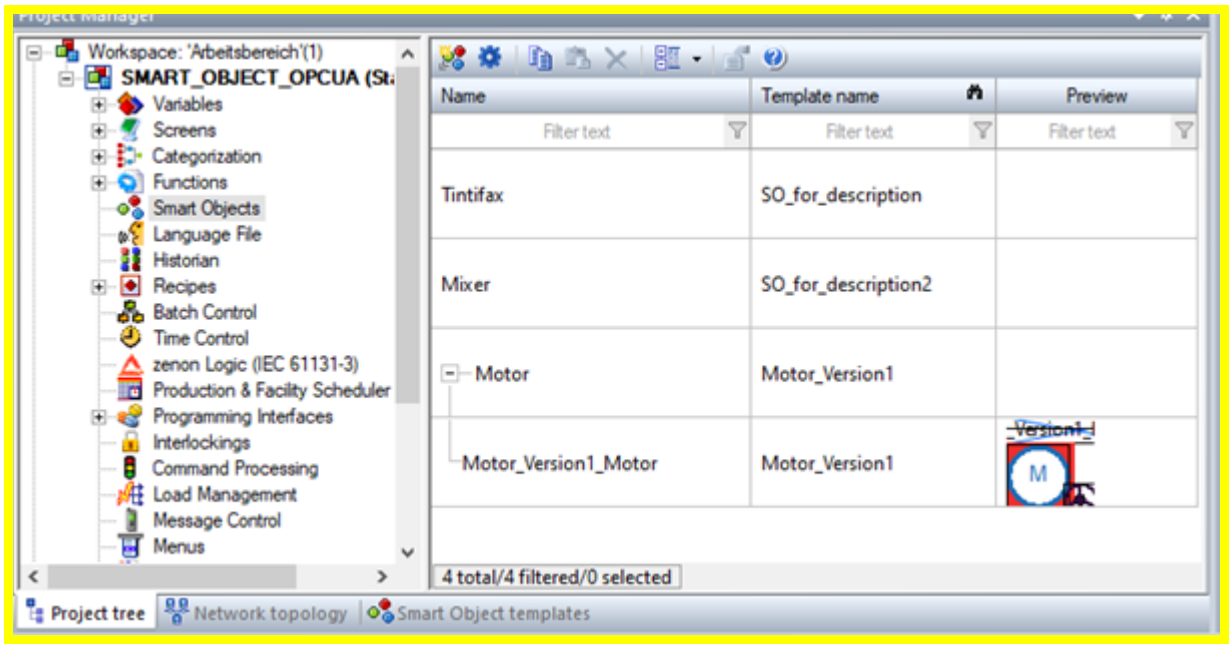
After a smart object is created, the configuration contents are applied in the active zenon project. If a smart object contains, for example, symbols and functions, these are created accordingly in the project in the project tree.

USE SMART OBJECT CONTENTS IN THE CURRENT PROJECT

The following applies for the use of smart object contents:

- Variables of the current project can be mapped in a dialog to the variables configured in the smart object.
  To do this, click the **Variable mapping** smart object property.
- Released properties can be parameterized by the user of the smart object.
- Properties that have not been released apply the parameterization of the smart object template.

6.1.2.1 Smart object symbols



The following applies for smart object symbols:

- A smart object symbol can be used in a zenon screen. To do this, drag the symbol from the smart objects list and drop it onto a zenon screen. No further configurations of the symbol

(such as modifying the linking rules) are necessary because these have already been defined automatically during the creation of the smart object.

- The smart object symbol is saved and displayed in the symbol library of the current project. This symbol can only be changed in the project via the released properties. The symbol is configured exclusively in the smart object template.
- It is possible to drag a smart object symbol from the symbol library and drop it onto a zenon screen, but this is not recommended. The contents (such as variables, functions, etc.) of smart object symbols that have been placed directly from the symbol library are not substituted!
- Manual editing of the linking rule of the symbol is not recommended. When the smart object is updated, such manual modifications are always overwritten!

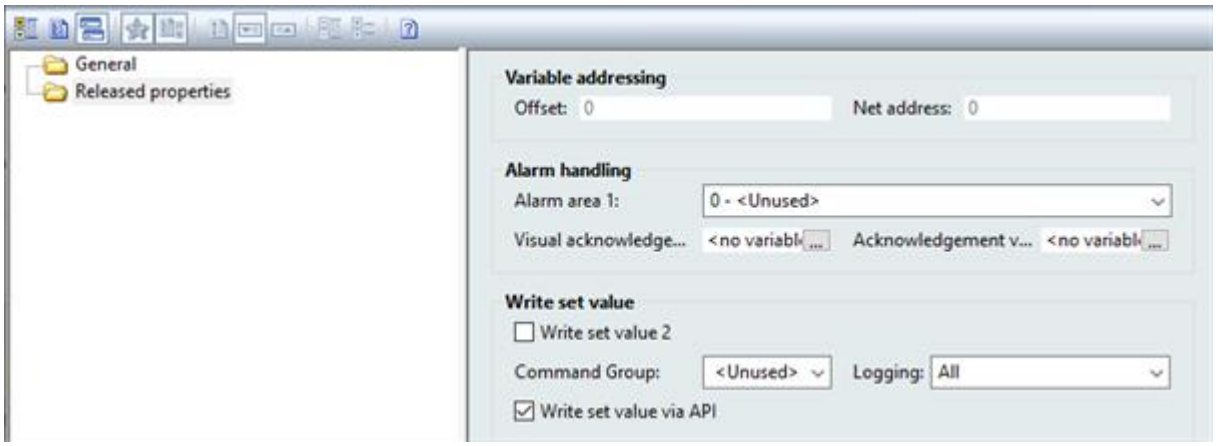
Information

If a symbol is not visible in a smart object, this can be due to the following reason:

The author of a smart object template can use the **Wird in der Smart Object Liste dargestellt** property to configure whether the smart object symbol is visible for the user. If this property has not been activated during configuration, the symbol is not visible as an individual component of the smart object and also cannot be used.

6.1.2.2 Edit released properties

The user of a smart object has the possibility to parameterize selected properties of previously created objects. This functionality allows you to customize all the created smart objects. In this way, a smart object can be used several times and with customized parameterizations.



In doing so, the following applies:

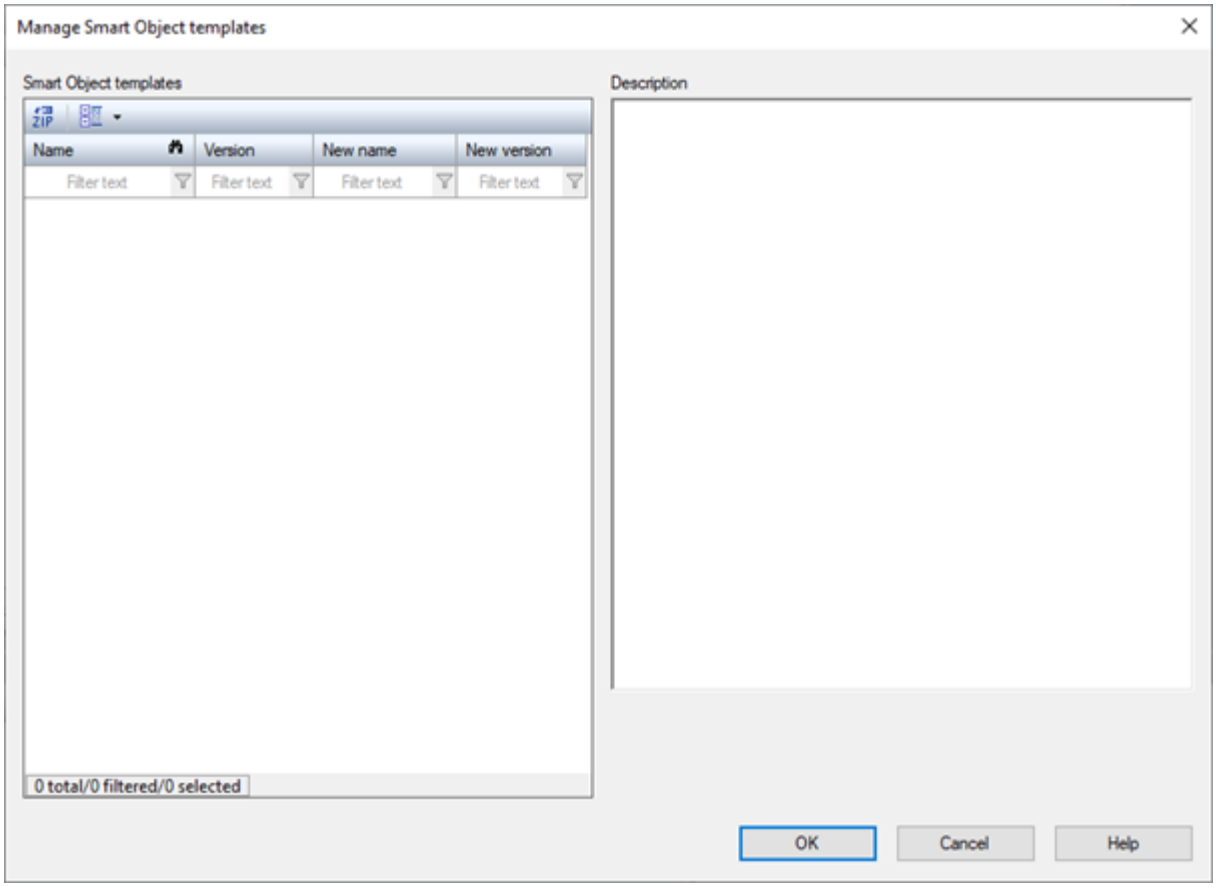
- Only those properties can be parameterized which have been released for parameterization by the creator of the smart object template.
- After a smart object is created, the released properties are available in the properties window.
- Properties that have not been released are write-protected and apply the parameterization of the smart object template.
- The names of the released property groups and properties are defined by the creator of the smart object template.
- Properties of the smart object element, such as the name of the smart object, are parameterized in the **General** smart object properties group,

The released properties can be linked or freely parameterized with objects of the current zenon configuration. This always requires that the corresponding property has been released during the configuration of the smart object template.
Examples:

- Linking of the alarm/event class of the current zenon project with the alarm/event class linked in the smart object template.
- Input of a limit value text.
- Activation of a required alarm acknowledgment.
- Background color of a screen.
- ...

6.1.3 Import smart object template

Follow the following steps to import a smart object template for the creation of a smart object:



1. Switch to the **Project tree** tab in the project manager.
2. Select the **Smart Objects** entry in the project tree.
3. Import a smart object template.
   a) To do this, click the symbol or the **Manage smart object templates** entry in the context menu.

      The **Manage smart object templates** dialog is opened.

   b) Click the **Import** symbol.

      This opens the file selection dialog.

   c) From the file list, select the .SO storage file for the **Smart object template** to be imported.
      Note: Multiple selection is not possible.

   d) Confirm your selection by clicking the **Open** button.
4. The smart object template is displayed in the list of available smart object templates.

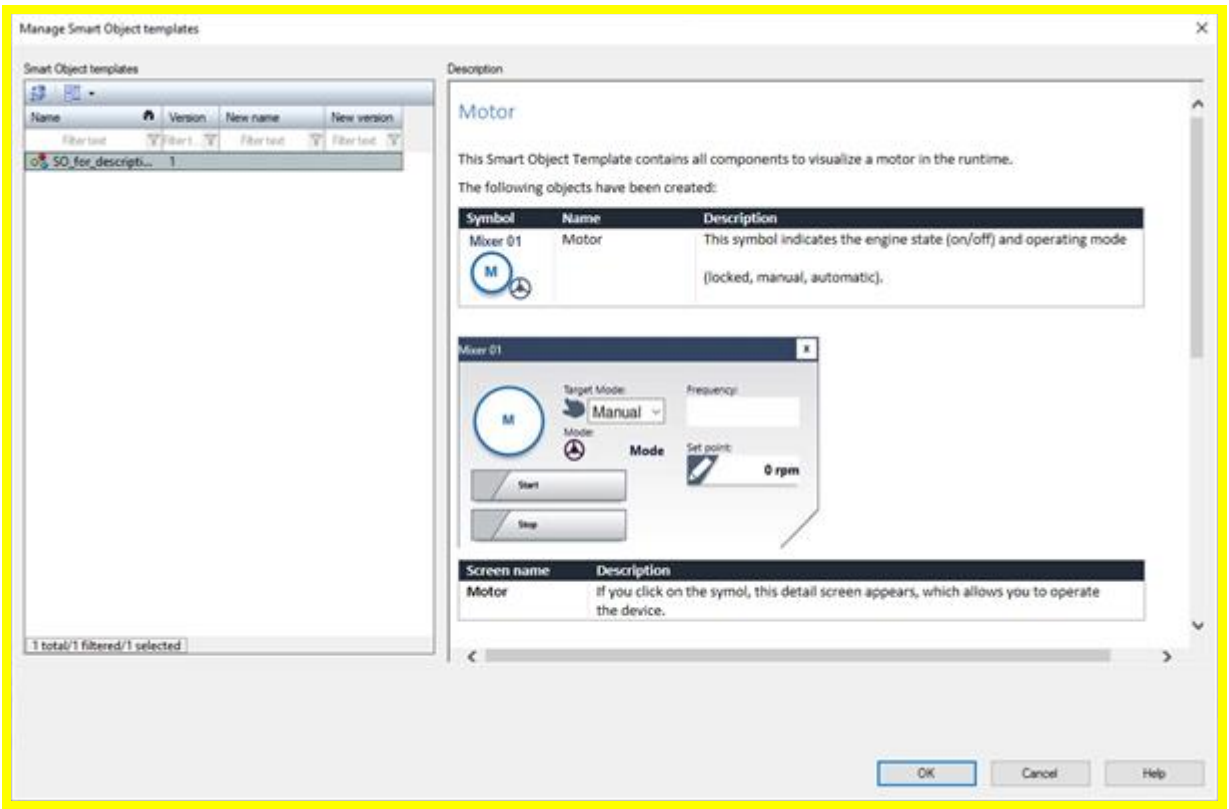
Note: If a smart object template is imported for the first time into the zenon project, the **Name** and **Version** columns are empty. You can find the relevant information of the new smart object template in the **New name** and **New version** columns.

5. Confirm your import by clicking the **OK** button.
6. The dialog is closed and the smart object template can be selected for the creation of a new smart object.

6.1.4 Update an existing smart object

In practice, smart object templates are usually modified and further developed independently of the smart objects derived from them. Thus, for instance, further properties can be released by the creator of the smart object template. Enhancements of an existing smart object template with zenon objects (variables, buttons, functions, symbols, etc.) increase the scope of services of an existing smart object template.

These modification and enhancements can be applied to existing smart objects. To do this, the smart object template is imported into zenon in a more recent version. All smart objects are enhanced by the contents of the smart object template on which they are based. Smart objects that already exist in the project are updated via the **Manage smart object templates** dialog. The version in use and the current version are visualized during the import process.



Follow the following steps to re-import a smart object template:

1. Switch to the **Project tree** tab in the project manager.
2. Select the **Smart Objects** entry in the project tree.
3. Import a smart object template.
   a) To do this, click on the symbol or the **Manage smart object templates** entry in the context menu.
   The **Manage smart object templates** dialog is opened.
   b) Click the **Import** symbol.
   This opens the file selection dialog.
   c) From the file list, select the **.SO** storage file for the **Smart object template** to be imported.
   d) Confirm your selection by clicking the **Open** button.
4. The smart object template is displayed in the list of available smart object templates.

   Note: The **Name** and **Version** columns display the version currently in use (before the import). The **New name** and **New version** columns define the version that is used after import for all the smart objects based on this template.
5. Confirm your import by clicking the **OK** button.
6. The dialog is closed and the smart object template is imported. All the contents of smart objects based on this template are replaced by the contents of the currently imported smart object template.

NOTES ON PROJECT CONFIGURATION

- Project backup before import

  Create a project backup before importing a smart object template. The updating of the smart object template can lead to major changes in the project.

- Use the description of the smart object templates for differences in the versioning

  A comparison of different smart object template contents is not possible. To do this, use the description. Document the content modifications of the versions as description text.
  Example: Summarize your changes in a table or as short text on the HTML page of the description.

- Check the existing configurations

  Released properties for configurations that already existed before being updated via smart object template import must be checked.

- Older and newer versions

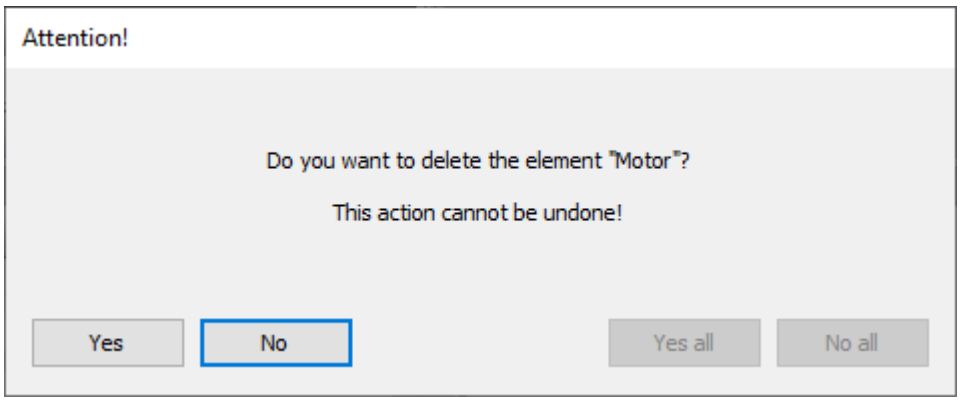
  It is possible to update both newer and older versions of smart object templates. This means that smart objects already existing in the project can also be updated by an older version of a smart object template. An import can lead to objects which have been created by the smart

object (such as variables, screens, functions, etc.) being removed from the project.
Example: If a smart object variable has been linked to an archive in the project, in this case, the variable is deleted from the project, the linking to the archive is thus invalid and is labeled accordingly.

Attention

When a smart object template is imported, any modified released properties of the current zenon project are overwritten with the content of the template!

6.1.5 Delete smart object



Follow the following steps to delete a smart object from the configuration:

1. Go to the **Smart Objects** node in the project tree.
2. In the detail view, click on the entry for the smart object you wish to delete.
3. In the toolbar, select the symbol or the **Delete** entry in the context menu.

   The confirmation dialog for deleting a smart object is opened.
4. Confirm the prompt by clicking on the **OK** button.
   The selected smart object is deleted.

Attention

When a smart object is deleted, all related zenon objects (such as screens, screen elements, variables, functions, etc.) are removed from the project. If these objects have been previously linked to other zenon modules, these links are no longer valid due to deleting the smart object. The broken links are labeled accordingly.

6.2 Configuration of smart object templates

Configuration contents and parameterizations are merged in a smart object template. Smart object templates are configured and parameterized in their own area in zenon. The engineering environment of the smart object template can be found in the **Smart Object templates** tab in the project manager.

In this section of the manual, you can find engineering instructions as well as tips for the creation, management and parameterization of smart object templates.

6.2.1 Create smart object templates

There are several options for creating a smart object template:

- Create manually (on page 45)

  The smart object template is newly created in the engineering environment of the smart object template.
- Create using the equipment model (on page 46)

  The smart object template is created from an existing configuration in the project tree.
- Import smart object templates from a file (on page 47)

  An already configured smart object template is applied by another project or another project engineer (for example, the systems integrator).

6.2.1.1 Manually create a smart object template

Follow the following steps to create a smart object template:

1. Switch to the **Smart Object templates** tab in the zenon project manager.
2. Create a new smart object template.
   In the tree, click the symbol or the **Create smart object** context menu entry. A new smart object template is created in the tree.
3. Click on the newly created object in the tree view. In the detail view of the smart object template, all the supported contents are, in turn, displayed as a tree structure. Brief information about the smart object template can be shown in the **Description** tab. However, this display is empty after a new smart object template is created.
   You can find further information on how to configure the description of a smart object template in the Create a description for a smart object template (on page 59) in this manual.
4. Now configure all the content that should be contained in the smart object template. The configuration is the same as in the zenon Editor.

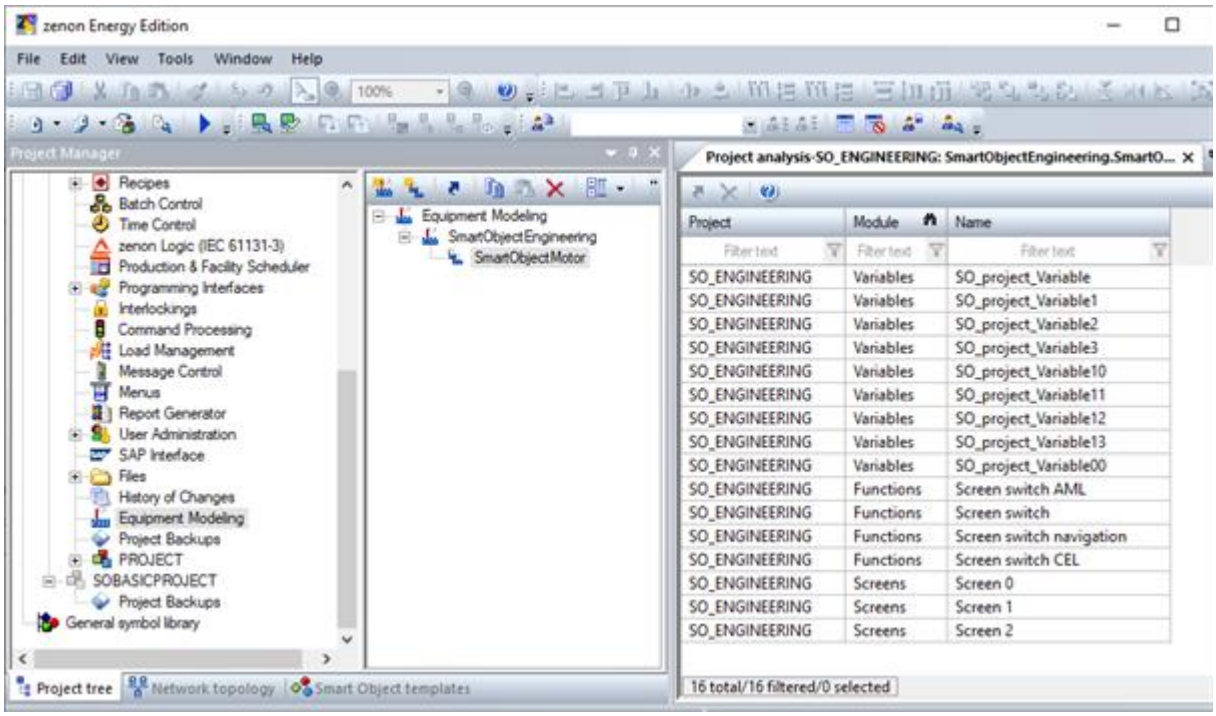
DUPLICATE AN EXISTING SMART OBJECT TEMPLATE

An existing smart object template can be duplicated. The copy can act as the basis for a new template. Follow the following steps to duplicate a smart object template:

1. Select the **Smart Object templates** tab in the project manager.
2. Select the desired smart object template in the tree view.
3. Click the **Copy** symbol or the corresponding entry in the context menu. The selected object is copied to the clipboard.
4. Go to the node in the tree view which should contain the new smart object template.
5. Click the **Paste** symbol or the corresponding entry in the context menu. The newly created smart object template is inserted at the selected position.
6. Give it a new name. To do this, parameterize the **Name** property in the properties window:
7. You can now edit the smart object template.

6.2.1.2 Create a smart object template using the equipment model

The Equipment Modeling module allows you to convert configurations from the zenon Editor into a smart object template with just a few clicks of your mouse. To do this, you must link all the elements of an existing zenon project that should be contained in the smart object template with a corresponding equipment group.



Follow the following steps to create a smart object template using the zenon equipment model:

1. Go to the project tree.
2. Create a new equipment group:
   - Go to the **Equipment Modeling** module in the project tree.
   - Configure a new equipment group.
     Tip: Create a separate equipment model for configurations of smart object templates. This allows you to use different equipment groups for the automatic creation of smart object templates. A distinct name makes mapping easier.
     Example: Give an equipment model the name *SmartObjectEngineering*. Then name the corresponding equipment group, for instance, *SmartObjectMotor* or *SmartObjectMixer*.
3. Link each element of your existing configuration that you wish to merge in a smart object template with the equipment group.
   Tip: In the project analysis, you can see an overview of all the elements that are linked with the equipment model. To do this, select the **Linked elements** entry in the context menu of the equipment group. The result is displayed in the main window.
4. Create a smart object template from the linked elements.
   To do this, select the **Create smart object** entry in the context menu of the equipment group.
   Attention: Please pay attention to the messages in the output window.
5. Switch to the Smart Object templates tab in the project manager in the zenon Editor.
   - You can now see the newly created smart object template in the tree view. The name of the template consists of the name of the equipment model and the equipment group (separated by a simple underscore).
   - The detail view contains the configuration of the zenon contents of the project linked with the equipment model.
   - This configuration can be modified and enhanced.

   Note: Changes to the configuration of the smart object template do not have an impact on the existing configuration in the project tree.

A new smart object template is created for each execution of the command. It is not possible to enhance an existing template by repeatedly executing the **Create smart object** command.

6.2.1.3 Import smart object template

Follow the following steps to import an smart object template:

1. Switch to the **Smart Object templates** tab in the zenon project manager.
2. Import a smart object template.
   a) To do this, click the main node or a configured folder in the tree view.
   b) Click the **Import** symbol.

This opens the **Manage smart object templates** (on page 11) dialog.

c) Click the **Import** symbol.

This opens the file selection dialog.

d) From the file list, select the **.SO** storage file for the **Smart object template** to be imported.
Note: Multiple selection is not possible.

e) Confirm your selection by clicking the **Open** button.

3. The smart object template is displayed in the list of smart object templates.

Note: If a smart object template is imported for the first time, the **Name** and **Version** columns are empty. You can find the relevant information of the new smart object template in the **New name** and **New version** columns.

4. Confirm your import by clicking the **OK** button.
5. The dialog is closed. The imported smart object template is displayed in the tree view of the smart object templates.

6.2.2 Edit smart object template

When a smart object template is configured, you can define the properties which should be parameterized by the user of the smart object.

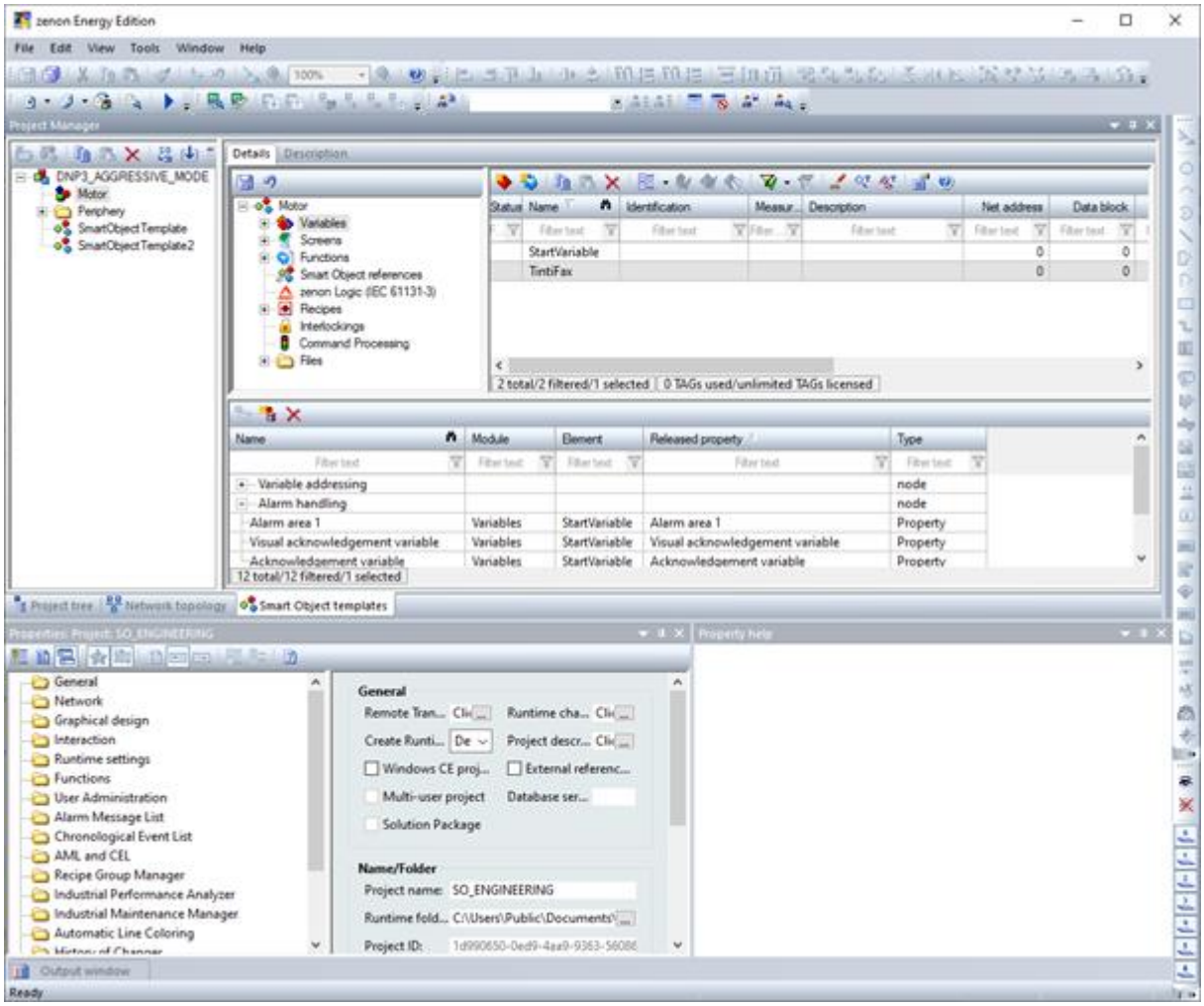
The goal should be to keep the complexity and the configuration effort as low as possible for the user of a smart object. This involves making only those properties available for parameterization that are absolutely necessary for use in the smart object. The released properties depend upon the expertise of the smart object user.

Therefore, the creator of a smart object template must carry out the following configuration steps:

- Release properties (on page 49)
  Configuration of the properties that are available to the user of a smart object.
- Merge properties (on page 52)
  Merging of several properties into one configurable property in the smart object.
- Structure properties (on page 54)
  Organization of properties in segments/groups.

6.2.2.1 Release properties

Released properties can be parameterized by the user in a smart object. Unreleased properties of a smart object can no longer be modified as a component of a smart object. These properties apply the parameterization of the smart object template.



Follow the following steps to release a property for parameterization in a smart object:

1. Switch to the engineering environment of the smart object templates.

   To do this, click the **Smart Object templates** tab in the project manager.

2. Select a smart object template in the tree view.
   This displays the engineering environment for the smart object template.

3. Select an entry such as **Variables** in the project tree of the selected smart object template.

4. Click in the detail view of a (previously created) variable of the smart object template.

   The variable and all its properties are displayed in the zenon Editor.

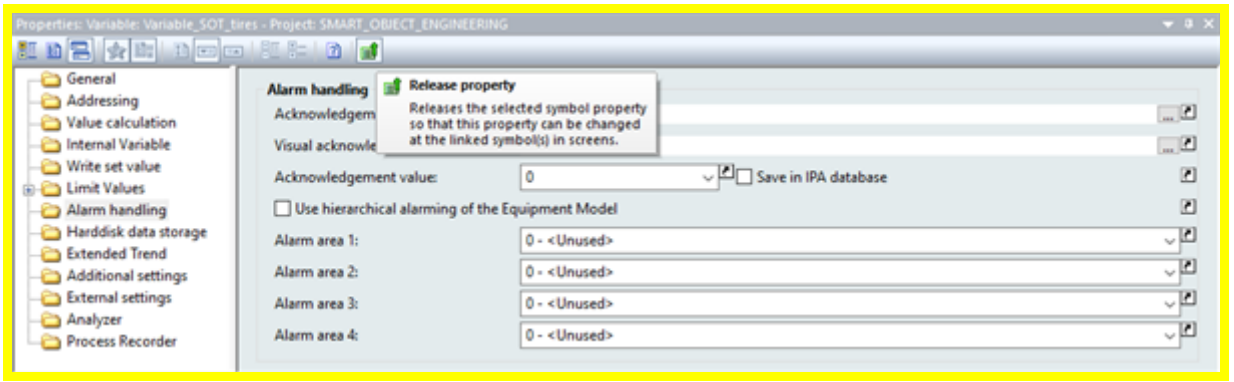
5. Parameterize the variable properties.

6. Release those properties that should be parameterized by the user of the smart object.

   a) To do this, click the desired property in the corresponding properties group in the properties window.

   b) Apply this property in the released properties area:

   ▸ To do this, drag the property name of the property onto the list of released properties of the smart object template.

   ▸ Click the property name and click the **Release property** symbol in the toolbar of the properties window.

   ▸ Open the context menu of the property to be released and select the **Release property** option.

   The selected property is applied in the list of released properties. The current parameterization is maintained. When it is used in the smart object, this applied parameterization is regarded as the default value for the property in the smart object.

7. Optional: Name the released property.

   a) To do so, click slowly on the entry in the **Name** column. The cell will be released for editing.

   b) Rename the released property.
   This name is displayed in the properties window of the smart object during parameterization.

Hint

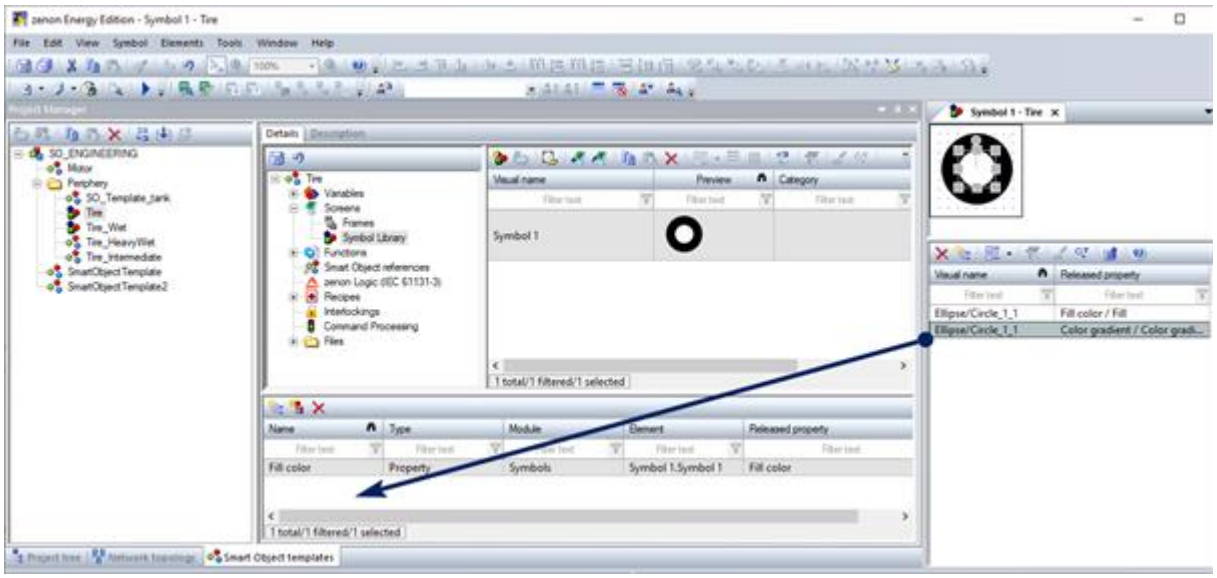
Thanks to the multiple selection feature in the detail list of the project tree of the smart object templates, you can apply the same property for all the highlighted elements with just one click.

Example: The configuration of a smart object template contains five variables.

- Highlight all five variables in the detail list.
- Click the **Identification** property in the properties window.
- Apply this property in the list of released properties.

The property is included separately in the list of released properties for each variable. This means that five entries for the released property are listed in the list of released properties: for each variable, the respective property.

RELEASE PROPERTIES FOR SYMBOLS



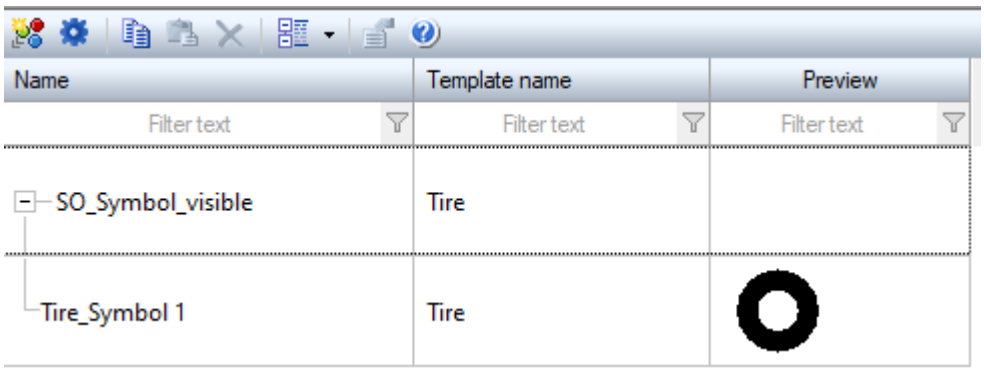
Properties of symbol elements (such as width and height of screen elements) cannot be released directly in the smart object template. These properties must first be applied in the released properties of the symbol. Afterwards, these released symbol properties can be dragged and dropped onto the released properties of the smart object template.

DISPLAY A SYMBOL IN THE SMART OBJECT

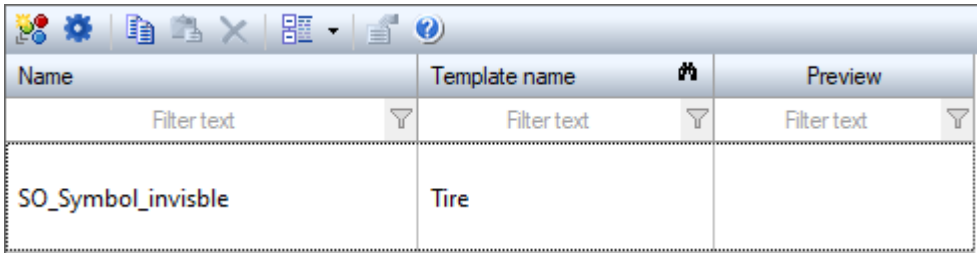
When parameterizing a symbol in a smart object template, you can define whether the symbol is visible to the user when it is used in a smart object. To do this, parameterize the corresponding **Wird in der Smart Object Liste dargestellt** property.

Released properties of a symbol are always visible and configurable for the user.

Example 1: Wird in der Smart Object Liste dargestellt property activated:



Example 2: Wird in der Smart Object Liste dargestellt property deactivated



6.2.2.2 Merge and combine released properties

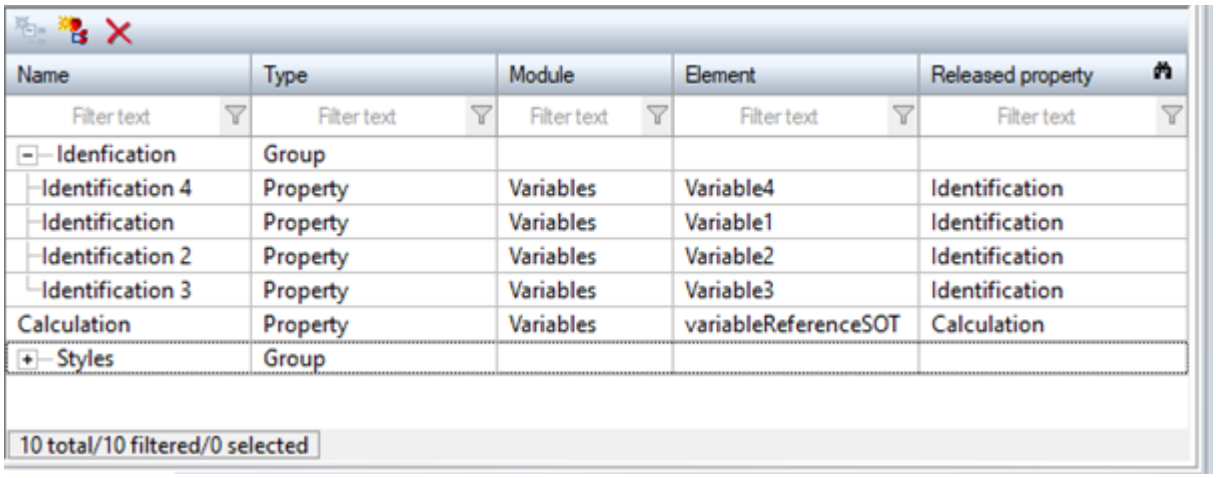
Several properties with the same configuration content can be merged for the configuration of a smart object. This means that only one property is available for the parameterization in the smart object. All the elements of the properties group are assigned the same value during parameterization.

When parameterizing a smart object template, this is done by creating a group. The created group can only contain properties with the same configuration content.

Example:

- Properties for the graphic display

- of a property of several zenon elements: **Identification** of several configured variables.



Name	Type	Module	Element	Released property
Filter text	Filter text	Filter text	Filter text	Filter text
Idenfication	Group			
Identification 4	Property	Variables	Variable4	Identification
Identification	Property	Variables	Variable1	Identification
Identification 2	Property	Variables	Variable2	Identification
Identification 3	Property	Variables	Variable3	Identification
Calculation	Property	Variables	variableReferenceSOT	Calculation
Styles	Group			

10 total/10 filtered/0 selected

Follow the following steps to merge several properties for the parameterization in a smart object into one group:

1. Switch to the engineering environment of the smart object templates.
   a) To do this, click the **Smart Object templates** tab in the project manager.
1. Select a smart object template in the tree view.
   This displays the engineering environment for the smart object template.
2. Select an entry such as Variables in the project tree of the selected smart object template.
3. Create a new group.
   a) To do this, select from the list of released properties the entry of a property you wish to merge with other properties in a group.
   b) In the toolbar, select the symbol or the **New group** entry in the context menu.
   A new entry is created in the list. This entry is named **Group** by default. The new entry is displayed as a node in the list and, as a sub-entry, contains the previously selected property.
4. You can drag and drop further properties to add them to the structure of the group. To do this, drag the selected entry onto the row of the group.

   Note: This means you can select several properties in the **List of released properties** and drag & drop them onto the group in one move.
5. Optional: Give the newly created group a name.
   a) To do so, click slowly the entry in the **Name** column. The cell will be released for editing.
   b) Rename the group.
   This name is displayed in the properties window of the smart object during parameterization.

> **Information**
>
> Released, merged properties groups (*Group* type) and individual properties can be organized in group boxes (*Node* type). In this way, a group can be a component of a node. However, a node can never be part of another node.

REMOVE ELEMENTS FROM THE GROUP - UNDO MERGE

Follow the following steps to remove one or several elements from the group of merged properties:

1. In the list of released properties, select the entry for an existing group.
2. Click on the [**+**] next to the entry. The entry is expanded.
   All the entries contained within are listed as subentries.
3. Drag and drop the entry onto the end of the list. The selected entry is removed from the group. Please pay attention to the mouse cursor during this step.

   Note: This means you can select several properties and use drag & drop to remove them from the group in one move.
4. If a configured group no longer contains any entries, the group will be deleted from the **List of released properties**

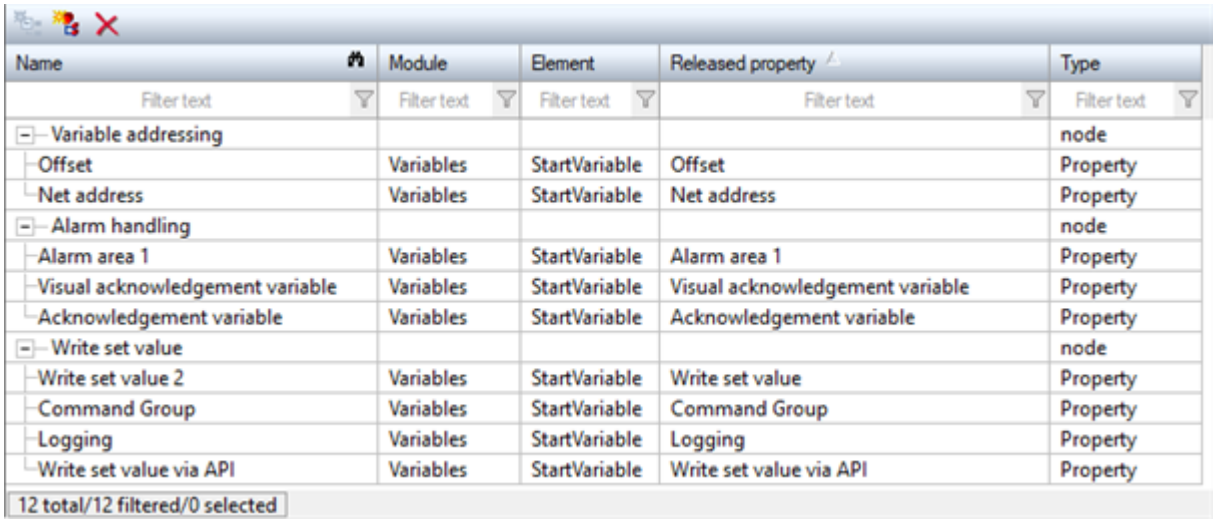
6.2.2.3 Structuring of released properties

Depending on the scope and use case, a smart object template can have a large number of released properties. To keep things manageable for the user, properties can be merged into nodes for display. Theses nodes are displayed in the smart object as a group box with all the released properties contained within it. The name of the group field corresponds with the name of the node in the list of released properties of the smart object template.

When structuring, related properties can be displayed separately in one node:

- All the properties that are parameterized for the configuration of a limit value.

- All the properties that should be parameterized by the user in the smart object for setting a limit value.



Name	Module	Element	Released property	Type
Filter text	Filter text	Filter text	Filter text	Filter text
Variable addressing				node
Offset	Variables	StartVariable	Offset	Property
Net address	Variables	StartVariable	Net address	Property
Alarm handling				node
Alarm area 1	Variables	StartVariable	Alarm area 1	Property
Visual acknowledgement variable	Variables	StartVariable	Visual acknowledgement variable	Property
Acknowledgement variable	Variables	StartVariable	Acknowledgement variable	Property
Write set value				node
Write set value 2	Variables	StartVariable	Write set value	Property
Command Group	Variables	StartVariable	Command Group	Property
Logging	Variables	StartVariable	Logging	Property
Write set value via API	Variables	StartVariable	Write set value via API	Property

12 total/12 filtered/0 selected

Follow the following steps to display several properties for the parameterization in a smart object as a group box:

1. Switch to the engineering environment of the smart object templates.
   a) To do this, click the **Smart Object templates** tab in the project manager.
1. Select a smart object template in the tree view.
   This displays the engineering environment for the smart object template.
2. Select an entry such as **Variables** in the project tree of the selected smart object template.
3. Create a new node.
   a) In the toolbar, select the symbol or the **New node** entry in the context menu.
      A new entry is created in the list. This entry is named *Group box* by default.
4. You can drag and drop further properties to add them to the structure of the node. To do this, drag the selected entry onto the row of the node.

   Note: This means you can select several properties in the **List of released properties** and drag & drop them onto the node in one move.
5. Optional: Give the node a name.
   a) To do so, click slowly the entry in the **Name** column. The cell will be released for editing.
   b) Rename the node.
   This name is displayed as a header in the properties window of the smart object during parameterization.

Information

Released, merged properties groups (*Group* type) and individual properties can be organized in group boxes (*Node* type). In this way, a group can be a component of a node. However, a node can never be part of another node.

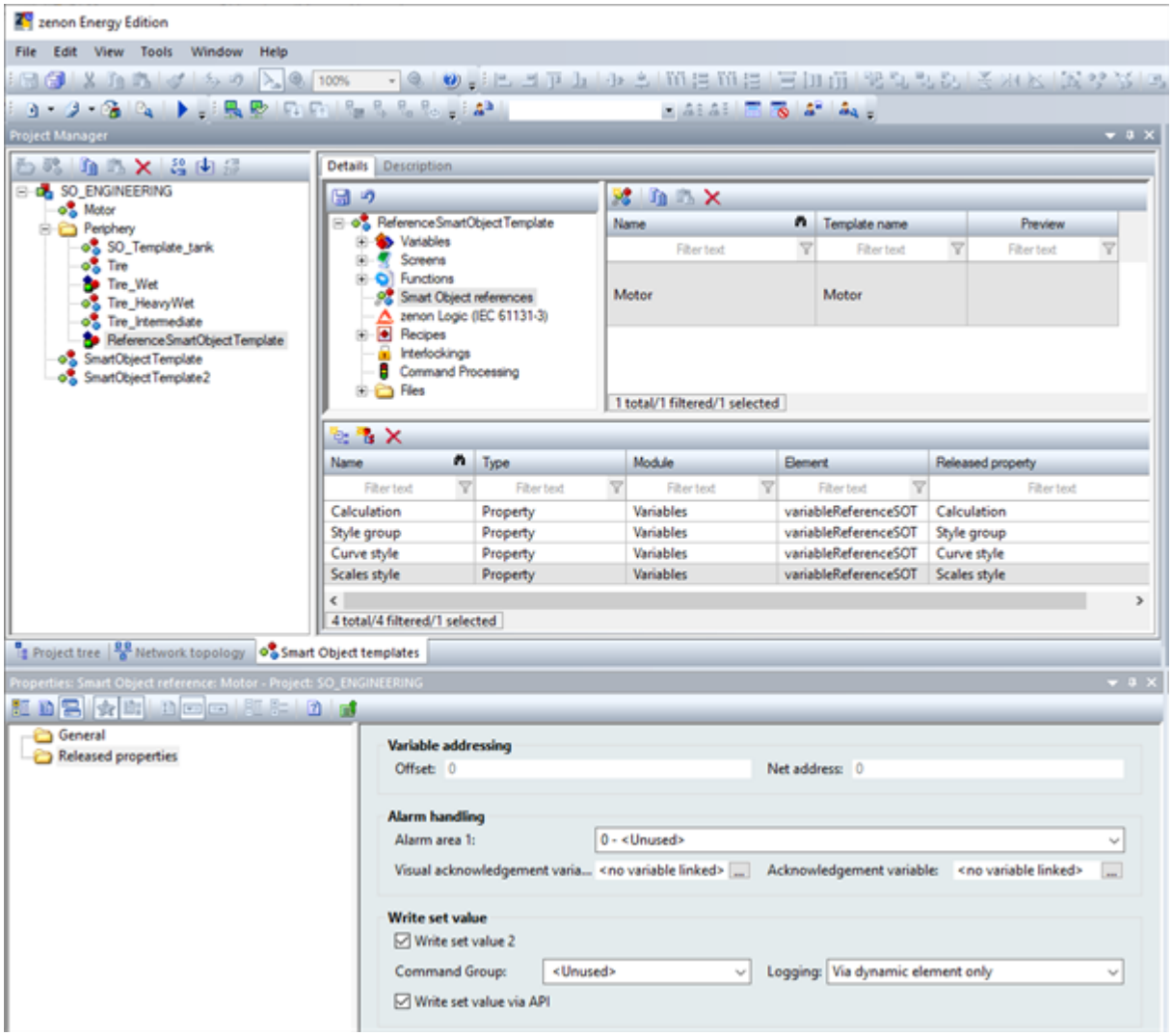
UNDO STRUCTURING

Follow the following steps to cancel the structuring for one or several properties:

1. In the **List of released properties**, select the entry for an existing node.
2. Click on the *[+]* next to the entry. The entry is expanded. All the entries contained within are listed as subentries.
3. Drag and drop the entry onto the end of the list. The selected entry is removed from the node. Please pay attention to the mouse cursor during this step.

   Note: This means you can select several properties and use drag & drop to remove them from the node in one move.
4. If a configured node no longer contains any entries, the entry is retained in the **List of released properties**.

6.2.3 Reference smart object templates



Follow the following steps to create a reference to an existing smart object template:

1. Switch to the engineering environment of the smart object templates.
   a) To do this, click the **Smart Object templates** tab in the project manager.
1. Select a smart object template in the tree view.
   This displays the engineering environment for the smart object template.
2. Select the **Smart Object references** entry in the tree view of the selected smart object template.
3. Create a new reference.
   a) In the toolbar, select the **Reference smart object** symbol or the **New reference** entry in the context menu.

   This opens the **Instantiate smart object** (on page 9) dialog.

b) Select an existing smart object template from the **Smart object templates** list as the basis for the reference.

c) Give the new reference to be created a name by entering a name in the **Name for new smart object** input field.

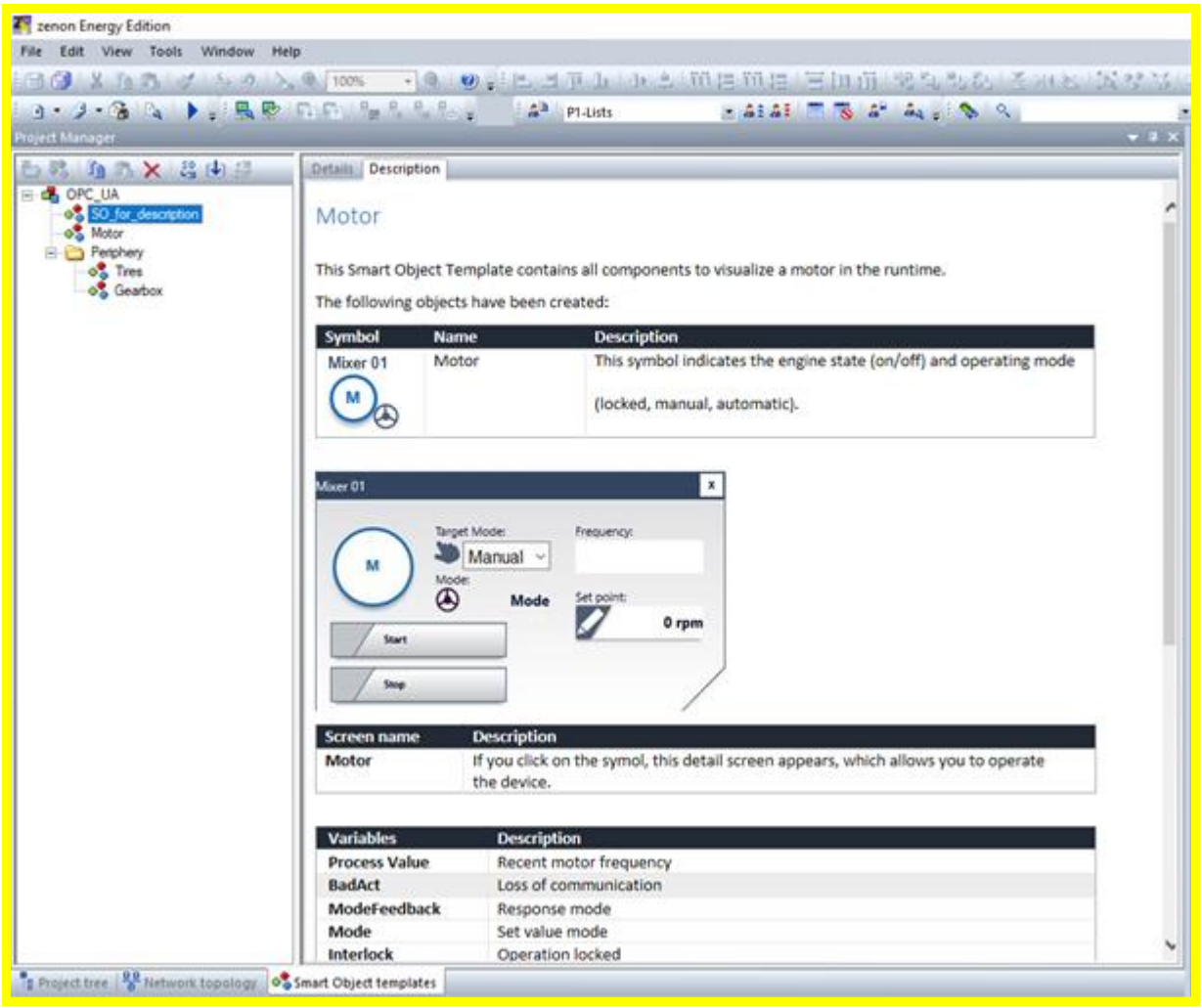
d) Confirm your configuration by clicking on the **Create** button.

4. The reference is displayed in the Smart Object references node in the detail view of the smart object template.

   Clicking on the reference displays the released properties in the properties window. This information is applied by the configuration of the linked smart object template.

> **Attention**
>
> The released properties of the referenced smart object template must be released again before they can be parameterized in the smart object. Released properties of referenced smart object templates are not automatically available for parametrization in the current smart object.

6.2.4 Create a description for a smart object template



In the case of a smart object template, a specific description can be saved in .HTML file format for the user. This information is displayed in the **Description** tab. The user will see this description when creating a smart object and in the management dialog. When a smart object template is created for the first time, it does not have a description.

When a smart object template is exported, an already existing description is included in the .so file. Thus, it is available automatically to the user when importing the template.

Follow the following steps to create a description for a smart object template:

1. Create a description for the smart object template in a .HTML file format.
   Attention: The (first) page must be named *index.htm*.

   For this, you can use any HTML editor, create simple HTML files from Word or create the HTML file in a text editor. Save your files.

2. Identify the GUID of the smart object template:

a) Switch to the **Smart Object templates** tab in the zenon project manager.

b) Select the smart object template in the project manager for which you want to add the description. In the properties, see the **GUID**.
Make a note of this GUID, for example, by taking a screenshot.

3. Open the storage location of the smart object template:

a) In the file directory, open the *C:\ProgramData\COPA-DATA\SQL2016\[Project GUID]\FILES\zenon\system\SmartObjects\[Smart Object Template GUID]\Description* folder.

Optional:

b) Select the active <CD_PROJECTNAME> project in the project tree. To do this, click the **Project tree** tab in the project manager and highlight the main node of the project.

c) Press the **Ctrl+Alt+E** keyboard shortcut. This opens the storage directory of the project.

d) Go to the *system* and *SmartObjects* subfolders. All the configured smart object templates are displayed in the file list. Each folder has the name of the **GUID** of the smart object template.

e) Open the *Description* folder within the smart object template folder.

4. Copy the previously created files for the information of the smart object templates in the corresponding language folder.

Each language folder represents the display language of the zenon Editor. This allows you to provide the user with information in the language in which the user is performing the configuration.
Tip: If the Editor is opened in Japanese, for example, and the respective language folder of the smart object template (*JAPANESE*) does not contain any content, the content of the *ENGLISH* folder will be displayed automatically in the Editor.

> **Hint**
>
> If you create a new version of an existing smart object template, a change history will be added to your description. This makes it easier to distinguish between different versions of a smart object template.

6.2.5 Make a smart object template available - export

Follow the following steps to export a smart object template:

1. Switch to the **Smart Object templates** tab in the zenon project manager.
2. Export a smart object template.

a) To do this, click on the main node or a configured folder in the tree view.

b) Select the smart object template to be exported in the project manager.

c) Click the **Export** symbol.

This opens the file selection dialog.

Note: Clicking on the symbol or the **Export all** entry in the context menu saves all the configured smart object templates in one file.

d) Select the saving location for the **.SO** storage file.

e) Confirm your selection by clicking on the **Save** button.

> **Information**
>
> If changes have been made in the smart object template, this changes the display of the smart object template icon in the project manager. Therefore, it is immediately obvious in which smart object template changes have been made.

VERSION NUMBER OF THE SMART OBJECT TEMPLATE

The following applies to versioning:

- Newly created smart object templates always have the version number *0**.
- When a smart object template is exported for the first time, it is given the version number *1*. If the template is exported again without changes having been made, the **Version** remains the same.
- If changes have been saved, but the template has not been exported, the version number remains the same and the current version number is indicated by a *. The template will not be updated to the next higher version until it is exported again.