

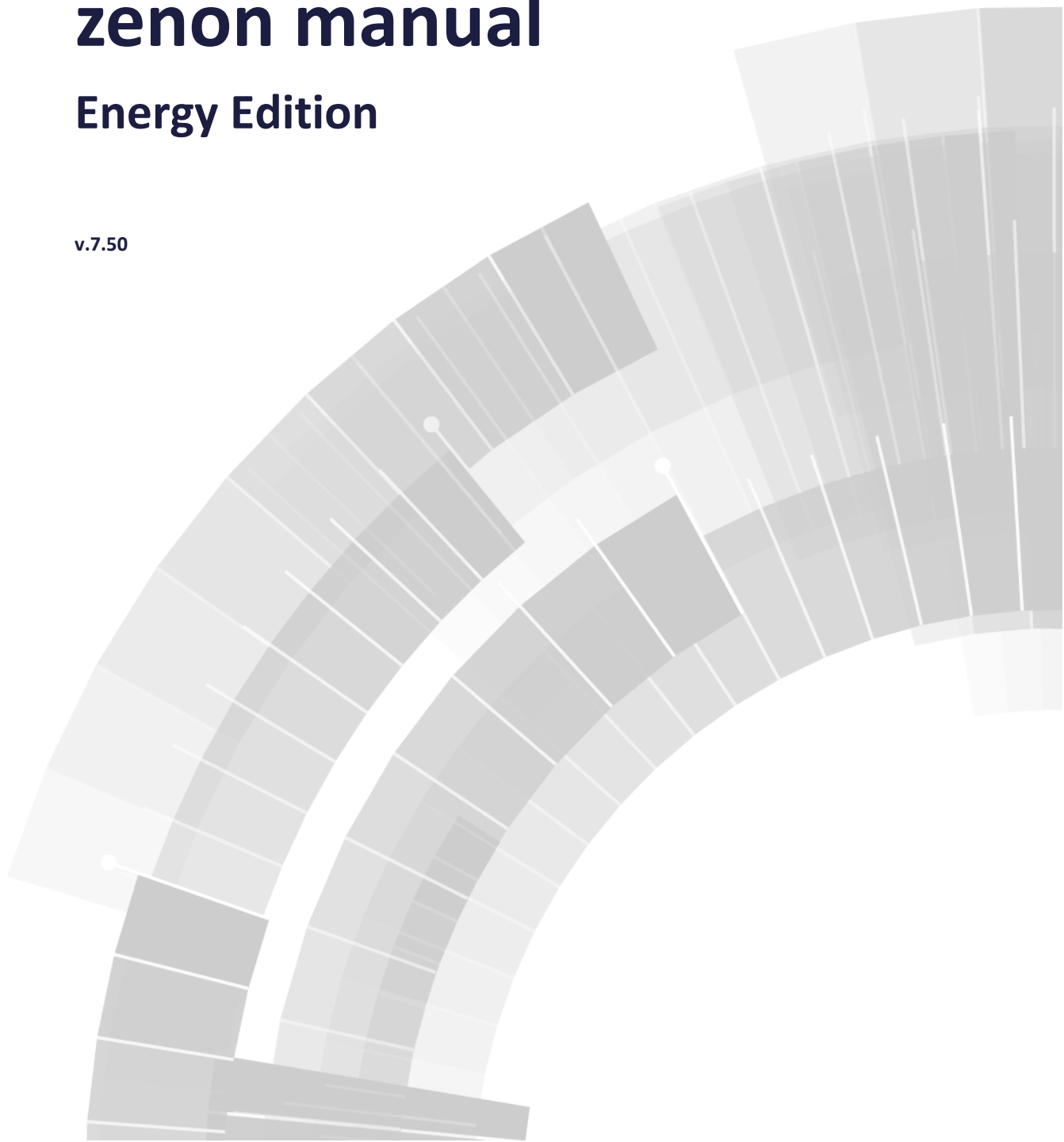


COPADATA
do it your way

zenon manual

Energy Edition

v.7.50





©2016 Ing. Punzenberger COPA-DATA GmbH

All rights reserved.

Distribution and/or reproduction of this document or parts thereof in any form are permitted solely with the written permission of the company COPA-DATA. Technical data is only used for product description and are not guaranteed qualities in the legal sense. Subject to change, technical or otherwise.

Table of contents

1. Welcome to COPA-DATA help	5
2. Energy Edition	6
3. Automatic Line Coloring (ALC) - Topology	7
3.1 ALC elements	8
3.1.1 Procedural elements	9
3.1.2 Lines.....	20
3.1.3 Checking the project.....	26
3.2 Configuration	27
3.2.1 Configuration of the sources	28
3.2.2 Configuration of topological interlockings	32
3.2.3 Configuration of the screen marker	36
3.3 Change ALC source color.....	37
3.4 Detail screens.....	39
3.5 Fault locating in electric grids	42
3.5.1 Search for ground fault.....	43
3.5.2 Short circuit search.....	50
3.6 Impedance-based error detection and calculation of load distribution	54
4. Command Processing.....	57
4.1 Command processing detail view toolbar and context menu	59
4.2 Engineering in the Editor	62
4.2.1 Creating a screen of the type Command Processing.....	62
4.2.2 Variables of the command group	65
4.2.3 Define Command Processing.....	68
4.2.4 Create menu	113
4.2.5 Create Runtime files - start the Runtime.....	115
4.3 Operating during Runtime	119
4.3.1 Command procedure.....	120
4.3.2 Command Processing screen control elements	128
4.3.3 Reload project online	160
4.3.4 Logging in the CEL.....	161

4.3.5	Lock return variable.....	161
4.3.6	Server change in redundant operation	161
4.3.7	Exit Runtime	162

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. Energy Edition

zenon Energy Edition is a package with special functionality for the energy sector and the procedural technology. The user benefits from easy-to-implement functions that allow for an individual adjustment of the application to the physical environment.



License information

Must be licensed in Editor and Runtime.

The following is available for the Energy Edition:

- ▶ Command Processing
- ▶ ALC (Automatic Line Coloring): Already included in the license for Energy Edition, provides basic properties for line coloring.
- ▶ Command sequences
- ▶ Topological element transformer
- ▶ Topology package: Requires additional licensing on the server (not on the client) and expands ALC by:
 - Multiple supply
 - Secured supply
 - Topological interlockings
 - Topological element disconnecter
 - Error detection and ground fault search

3. Automatic Line Coloring (ALC) - Topology

The topological coloring of lines allows easy automatic dynamizing of tubes in technology (for media) as well as in the energy distribution (for electricity). So process controlled coloring of topological nets can easily be realized.

Because the tube structure is designed in the screen with all its technological elements (e.g. tanks and valves, or generators, switches and consumers), it is internally emulated as a model and the media flow is displayed in the Runtime.

In order to allow screen-overlapping models the entire design and configuration is always project-wide. You therefore have one entire topological model per project, which is used for the calculation of the tube statuses and ultimately for the coloring of the tubes.

The whole topology is created automatically from the graphic design. No other engineering actions are necessary.



Information

The ALC algorithm only runs through once from a source starting from each switch.

DETAIL SCREENS

To display individual screens, a partial area can be taken from the topological network and displayed individually by means of alias. A detail screen (on page 39) can be displayed with the data from different equipment parts, for instance outputs or partial networks.



License information

Must be licensed for Editor and Runtime (single-user, server, standby).

No need to be licensed for Runtime client.

Licensing is carried out using the zenon Energy Edition.

- ▶ ALC: Included in the license for Energy Edition; provides basic properties for line coloring and multiple supply.
- ▶ Topology package: Requires additional licensing on the server (not on the client) and expands ALC by:
 - Secured supply
 - Topological interlockings
 - Transformer and separator topological elements
 - Error detection (version 6.50 and above)

3.1 ALC elements

Automatic Line Coloring (ALC) makes it possible to color lines regardless of the process status. The combined element is used as the process element. Automatic line coloring allows easy automatic dynamizing of tubes in technology (for media) as well as in the topological networks (for electricity).

ENGINEERING

For the design two types of screen elements with different functions are distinguished. On the one hand these are procedural elements (on page 9) (source, switch/disconnector, drain, transformer or link) and on the other hand lines (on page 20).

In doing so, the technical elements have a function and a color (source and transformer). If the procedural elements are active, the connected lines take on the color of these elements at the source and transformer or they take on the color of the element's input line for the switch and the link. If the procedural elements are inactive, the color of the lines is taken from the definition in the editor.

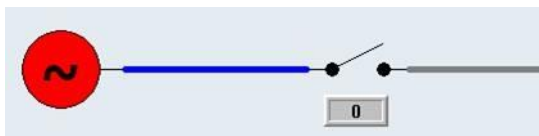
The different functions of the elements are assigned in the properties of the combined element.

EXAMPLE

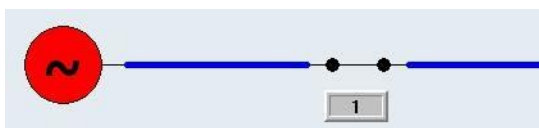
A source has a connected line. A switch is connected to the line. And a second line is connected there. If the source is active, the first line is colored with the color of the Automatic Line Coloring defined in the source up to the valve. The other line is not colored before the switch is closed.



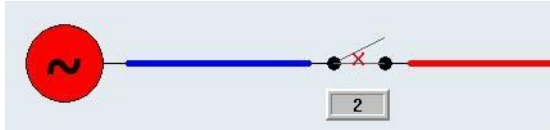
Source inactive



Source active



Switch closed



Undefined or invalid



Information

If the procedural element status is undefined or malfunction, this is automatically detected. All connected lines and all further elements are displayed in the color of the predefined source 'undefined' for both states.

NUMBER OF CLOSED SWITCHES IN A SERIES

For the correct functioning of the ALC algorithm, the number of connected switches in a series plays a role.

Recommendation: Arrange a maximum of 256 closed switches in a series between the source and the drain.

3.1.1 Procedural elements

Procedural elements are created in zenon with a combined Element. Their state determines the coloring of the connected line.

The following settings are available:

Property	Description
Function type	Defines the technological type of the Combined element.
Terminator	For bus bar ends. Blocks the error message " Line only connected on one side " when being compiled in the Editor.
Generator	A generator generally behaves like a source, but it is considered as an independent and not net-synchronous.
No function	The element has no function in the ALC. Note: The "no function" function type is the default value.
Link	<p>With a link a line can be continued on some other place. If a link is supplied by a line, all other links with the same link number also are supplied by this line. Here it does not matter, whether the links are in the same screen or on different screens in the project. So screen independent lines can be defined. It is possible to have more than two links with the same link number in one project.</p> <p>Links can be supplied by several lines at the same time or can themselves supply several lines. In principle there is no difference between inputs and outputs. The source information is passed on to all connected lines.</p> <p>Attention: Two link elements cannot be connected directly to one line. In between, there has to be at least one other procedural element (switch/disconnector or transformer).</p> <p>A link cannot be switched active or inactive: it always is active.</p>
Source	<p>Passes on its color. If the source is active (value: 1), all connected lines that have Color from ALC option set in the element properties are allocated the color of the source. The color is defined in the project properties as the source color. (e.g. tanks or generators). A source is a single pole with a static source number assigned to it. The source is switchable over the state of its main variable. Generally, sources are considered as net-synchronous and detachable.</p> <p>You can find details on the source in the configuration of the sources (on page 28) chapter.</p>
Switch	<p>With this lines can be split. If the switch is closed/active (value: 1), then the connection between the two lines is closed and the line is colored up to the next switch with the defined source color. In this case a switch forwards the source color of the input line to the output line.</p> <p>If the status of the switch is malfunction, undefined or INVALID, the color of the line turns into the color undefined from the ALC configuration in the project properties. A switch thus delivers source number 0 (undefined) to its output (connection 2) instead of the incoming source number.</p> <p>Example: see Switch example - colors from ALC (on page 14) section.</p> <p>Note: If the Switch input/output property is active, the input and output of this element are reversed for the ALC.</p>
Valve	<p>A slider (a valve) acts in a similar manner to a switch, but it is used for water and gas lines.</p> <p>Value of the main variable:</p>

	<p>Switch OFF: Value 0 -> Slider closed-> No forwarding</p> <p>Slider ON: Value 1 -> Slider open completely-> Water flow</p> <p>Slider DIF: Value 2 -> Slider partially open-> Water flow</p> <p>Slider STO: Value 3 -> Slider malfunction</p> <p>Note: If the Switch input/output property is active, the input and output of this element are reversed for the ALC.</p>
Drain	<p>This defines the end of the line. The drain does not influence the coloring; it is only used so that the model can be displayed in full. If an external program (e.g. VBA) should access the model, then the drain probably is needed for further calculations, and so has to be inserted.</p> <p>In Energy projects, the drain is used for representing consumers. These customers are considered for the calculation of the ALC interlockings (command groups) 'Consumer is undersupplied'.</p>
Check valve:	<p>The <code>check valve</code> only forwards information in one direction:</p> <ul style="list-style-type: none"> ▶ Value 0: The forwarding is not active (= closed) ▶ Value 1 or 2: Forwarding is only possible in one direction. In doing so, only forwarding from the input to the output takes place. Forwarding in the opposite direction is not envisaged. This also affects the ground switching. ▶ Value 3: Forwarding is undefined. This then occurs, for example if the <code>check valve</code> is faulty. In this case the status is only forwarded at the output. <p>Note: If the Switch input/output property is active, the input and output of this element are reversed for the ALC.</p> <p>The <code>check valve</code> is also taken into account by the topological interlockings (on page 32).</p>

Transformer	<p>A transformer is a drain and a source at the same time. SO with a transformer the input color (input source) can be transformed to a new output color (transformer source color).</p> <p>The output line is only switched to active once the transformer has an active input line. However the output line does not get the color of the input line as with a switch, but instead the color of the transformer's own source. So a source has to be defined for each transformer. A transformer cannot be switched active or inactive, it always is active.</p> <p>Note: If the Switch input/output property is active, the input and output of this element are reversed for the ALC.</p> <p>Reverse-feed-compatible transformer:</p> <p>To have a transformer capable of reverse feed, you must select, for Source for reverse feed, a different source than <code>UNDEFINED [0]</code>. This means that the transformer behaves the same for both directions - from the input to the output (forward) and also from the output to the input (backward). The only difference is that the Source for reverse feed property and not the Source property is used for further distribution of the source number.</p> <p>Note: Defective network statuses or missing configurations, such as a feed from the input and output at the same time or a short circuit from input and output are not specially colored. This means that the transformer capable of taking a reverse feed behaves like two transformers switched to run antiparallel that are not capable of taking a reverse feed.</p>
Disconnecter	<p>A disconnecter generally behaves like a <code>switch</code>. Nevertheless, a disconnecter may not be connected in the topological model. A status (<code>on</code>, <code>off</code>, <code>intermediate position</code>, <code>malfunction</code>) is determined via its main variable.</p> <p>Note: If the Switch input/output property is active, the input and output of this element are reversed for the ALC.</p>

SOURCE AND LINK NUMBER

Parameters	Description
Source	<p>Here a source is assigned to an element. In this selection box all sources defined in the ALC configuration (in the project properties) are available. All source names are listed.</p> <p>This property is only active if the function type 'source', 'transformer' or 'generator' has been selected.</p> <p>You can find details on the source in the configuration of the sources (on page 28) chapter.</p>

Link number	Only the link number is entered for a <code>link</code> function. All identical link numbers in a project correlate with each other. Detailed description in the function type Link. This property is only active, if the function type link has been selected.
--------------------	---

A variable of the IEC type BOOL or integer has to be linked to the element as the main variable, so that the switch can get the status (open, closed, invalid). In the same way, the source gets its status (active/inactive) from the linked main variable.

For the function types source and transformer the defined source number is forwarded to the consumers (drains) over open or closed switches. The statuses and colors of all connected lines are calculated from the superposed sum of the supplying source numbers and procedural elements.



Information

Only the first two bits are considered for the switching. The first bit stands for the actual switching. 0 equals off and 1 equals 1.

The second bit is the error bit. There is no error only if it is 0.

STATES

- ▶ A switch and a source are switched on if the value of the linked variable is 1.
- ▶ A switch is invalid if the value of the linked variable is >1 or has an `INVALID` status. An invalid switch provides the source number 0 (undefined) at its exit (connection 2) instead of the source number entering. In the direction towards the input the switch behaves as normal.

Note: if the (acknowledgment) variable has the status `INVALID`, the whole subsequent network is `INVALID`, because the status of the network is not known. The status `INVALID` is forwarded (routed) using subsequent closed switches.



Attention

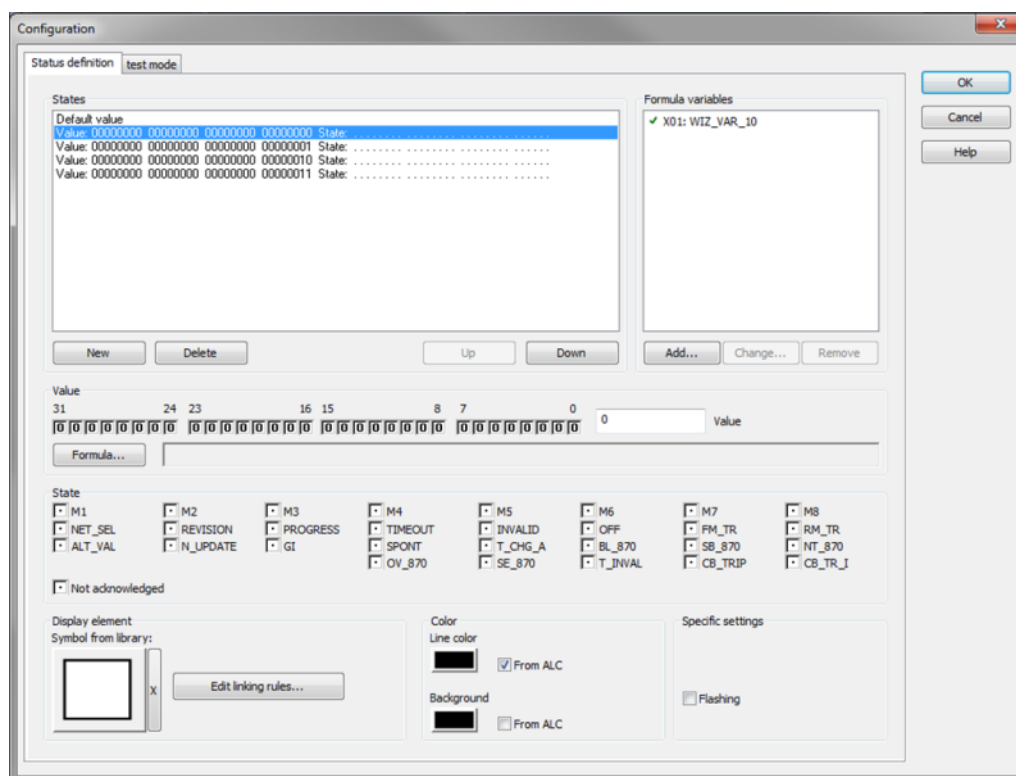
If in the single status the color and the fill color from the ALC is activated, also the procedural elements are colored by the status of the connected lines in the Runtime.

Switch example - colors from ALC

EXAMPLE 1

Combined element with value status 00 and line color from ALC:

1. Configuration in the Editor:
 - Combined element with value status 00
 - Line color from ALC active



2. Results in the following in Runtime:
 - Source color: green
 - Color without voltage: white
 - Switch status: off/open (value 0)

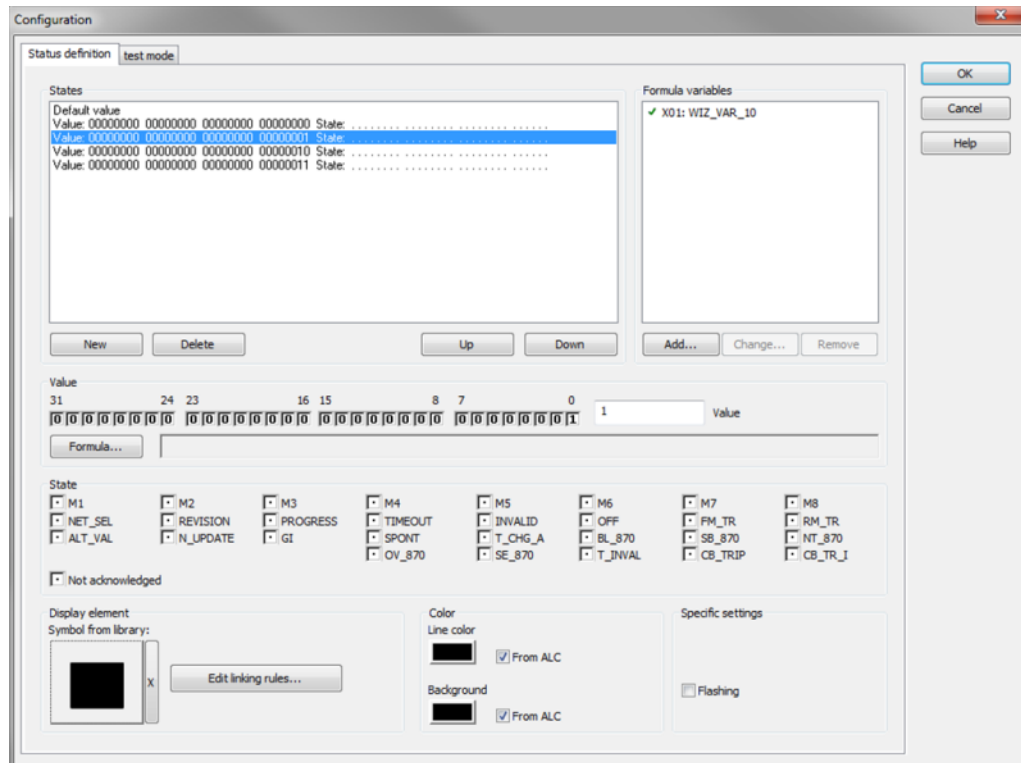


EXAMPLE 2

Combined element with value status 01 and colors from ALC:

1. Engineering in the Editor

- Combined element with value status 01
- Line color from ALC active
- Fill color from ALC active



2. Results in the following in Runtime:

- Source color: Green
- Color without voltage: White
- Switch status: on/closed (value1)

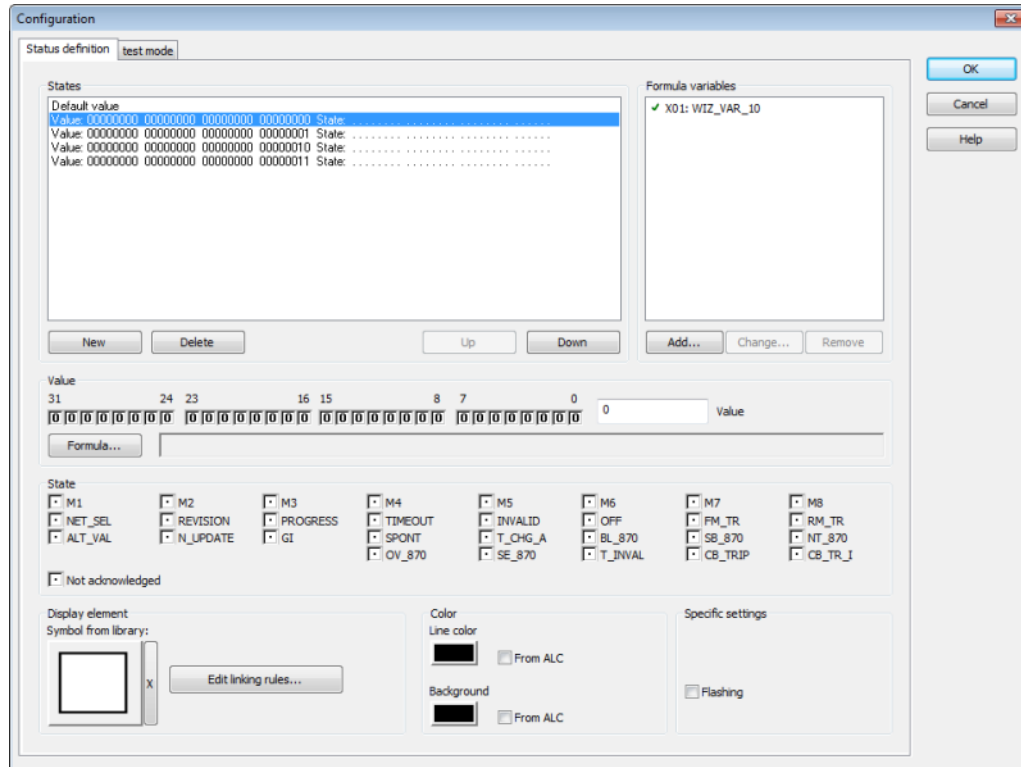


EXAMPLE 3

Combined element with value status 00 without colors from ALC:

1. Configuration in the Editor:
 - Combined element with value status 00

- Line color from ALC not active



2. Results in the following in Runtime:

- Source color: Green
- Color not energized and construction color of the line: White
- Defined line and fill color of the combined element: black
- Switch status: off/open (value 0)

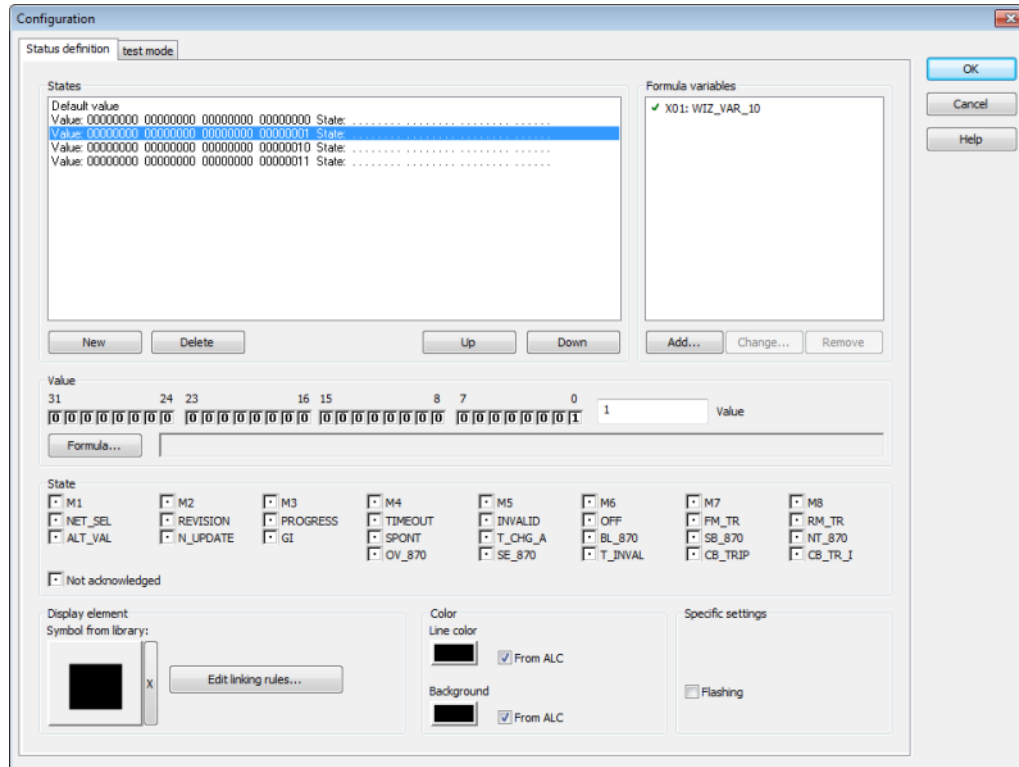


EXAMPLE 4

Combined element with value status 01 without colors from ALC:

- Engineering in the Editor
 - Combined element with value status 01
 - Line color from ALC inactive

- Fill color from ALC inactive



2. Results in the following Runtime:

- Source color = green
- Color not energized and construction color of the line: white
- Defined line and fill color of the combined element: black
- Switch status: on/closed (value1)



Connection points of procedural elements

When configuring, a line is connected to a procedural element (combined element) by overlapping drawings in the screen at connection points of the combined element. Only one line can be connected to the same connection point at the same time. All lines that start within the area defined, are connected (Topology from the graphic).



Attention

Use ALC elements only in un-rotated state because:

The calculation for the topological model for the ALC in the Editor is based on the position of the elements in un-rotated state and without considering any dynamics.

CONNECTION POINTS AND CONNECTION AREAS

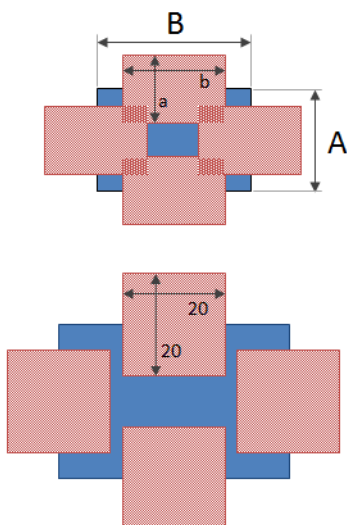
- ▶ The connection area for a connection point is in the middle of each side of the combined element. Each combined element thus has four connection points.
- ▶ The size of a connection area corresponds to $\frac{2}{3}$ of the height and width of a combined element, but no more than 20 pixels.
- ▶ Each connection area is centered in the middle of the respective element corner and stretches symmetrically inwards and outwards, to a maximum of 10 respective pixels.



Attention

If the combined element is less than 30 pixels, connection areas within an element overlap. Lines that could touch can cause errors (compilation, coloring).

You can see the possible connection points for combined elements smaller and larger than 30 pixels in the illustration.



Colors

- ▶ Blue: Combined element
- ▶ Red: Connection areas

Dimensions:

- ▶ **A**: height of the Combined element
- ▶ **B**: width of the Combined element
- ▶ **a**: Width of the connection area: $\frac{2}{3}$ of **A**, but a maximum of 20 pixels.
- ▶ **b**: Length of the connection area: $\frac{2}{3}$ of **B**, but a maximum of 20 pixels.

RULES

- ▶ If a line is outside the connection area, no connection is detected and there is thus no coloring of the line. So there will also be no coloring for further lines.
- ▶ With sources, drains and **Links**, all described connection points can in principle be used.
Attention: With sources and drains, only one connection point can be used at the same time. If different connection points are used at the same time, undefined states can occur.
 Elements of the type **Link** can also use several connection points at the same time. The incoming color information is passed on to all lines.
- ▶ With switches/disconnectors/sliders and transformers, the connection 1 (supply) is on the left or on the top and connection 2 (output) is on the right or on the bottom. This sequence can be changed with the **Switch input/output** property.
Attention: At switches and transformers it has to be cared, that only one input connection and one output connection is used. The simultaneous use of several input or output connection points results in inconsistencies and is therefore not reliable.
- ▶ For all procedural elements the following is true: Only one line can be connected to a connection point. Junctions cannot be realized directly on an element but must be drawn with lines.

Switch input/output

If a transformer, a disconnector or a switch is configured, the input and output can be swapped. To do this:

1. Select either transformer, disconnector or switch as a **Function type**
2. Activate the checkbox **Switch input/output**

The input is then set at the bottom right and the output at the top left.

OVERVIEW

Device configuration	Input	Output
normal	Left	Right
normal	top	bottom
swapped	Right	Left
swapped	bottom	top

3.1.2 Lines

Lines are represented by vector elements Line, Polylines and Tube.

If the option **Color from ALC** is activated for a line, the coloring is defined by the ALC configuration. Lines are automatically colored by the system depending on the status of the procedural elements and the ALC settings.

Here the color usually comes from the highest priority source number of the media flowing through the line, or stays "empty/not energized" just as defined in the screen with static or dynamic colors.

You define the display type by means of drop-down lists:

- ▶ Priority for display
- ▶ Display multiple supply
- ▶ Display secured supply

The following options are available in the properties of the lines:

Parameters	Description
Color from ALC	Activates the automatic line coloring for these vector elements. That means: If the source for the line is active and all switches/valves leading from the source to the line are closed/open, the line is accordingly colored. If the line is fed by a single source, the defined source color is used for coloring the line. The line width is not changed.
Priority for display	Defines if <code>multiple supply</code> , <code>secured supply</code> or both are displayed. Default: <code>Multiple supply</code>
<code>Secured supply</code>	The element is displayed according to the rules of the secured supply. A line is then considered to have a secure supply if it is supplied by at least two different switches or transformers with a non-system source. System sources do not contribute to secured supply, but do not exclude it.
<code>Multiple supply</code>	The element is displayed according to the rules of the multiple supply. A line is considered to have multiple supplies if it is supplied by at least two different sources. In doing so, it does not matter if they are system or user sources and from which side the line is supplied by the sources.
<code>No priority</code>	The coloring rules for <code>multiple supply</code> and for <code>secured supply</code> are applied at the same time if both criteria are met. That means: If a line <ul style="list-style-type: none"> ▶ has <code>multiple supplies</code> and a <code>secured supply</code>, ▶ The priority is set to <code>No priority</code>, ▶ The display for <code>multiple supply</code> is set to <code>two sources with highest priority</code>, ▶ The display for <code>secured supply</code> is set to <code>double width</code>, Then the line is twice as wide and displayed as a dashed line in two colors.
display multiple supplies	Multiple supply means that a line is supplied by multiple sources at the same time. Here you can define how lines with multiple supply are displayed. Default: <code>highest priority source</code>

highest priority source	The line gets the color of the source with the highest priority. Note: Priorities correspond to the sequence chosen in the ALC configuration.
two highest priority sources	Applies for lines fed by two or more different sources. The two sources with the highest priorities define the coloring. The line is displayed with these two colors (dashed). The dash length can be changed using the Dashing length supplied multiple times property. System sources apply for multiple supplies just as with genuine sources and color lines in two colors if they are configured accordingly.
Alternative color	The color defined in the Alternative color property is used.
Dashing length supplied multiple times	Defines the dash length (in pixels) of lines, polylines or tubes for the dashed ALC coloring for two sources with the highest priority for display multiple supplies. <ul style="list-style-type: none"> ▶ Minimum: 0 (automatic dash length) ▶ Maximum: 32767 ▶ Default: 0
Alternative color	Alternative color for the ALC coloring of lines, polylines or tubes with multiple supplies.
display secured supply	Secured supply means that a line gets multiple supply from one source (parallel). Here you can define how 'secured supply' is displayed. A line is always displayed as having a <code>secure supply</code> if it is supplied by at least two switches with a genuine source (not system source). Default: <code>normal</code>
double width	Relevant for lines fed in parallel by the same source. If this is the case, the line is displayed with double the configured width. (Example: A line with line width 5 pixels is displayed with 10 pixels if secure-fed.) If this line is fed by two or more different sources (multi-supply), the line width does not change! The color is <code>always</code> defined by the source with the highest priority!
double brightness	Relevant for lines fed in parallel by the same source. The line is displayed with double the original brightness. If this line is fed by two or more different sources (multi-supply), the line color does not change! If this line is multi-fed from one source (secure supply), the line is

	<p>displayed with double the original brightness. Formula for the calculation of the double brightness:</p> <ol style="list-style-type: none"> 1. The defined RGB color is transformed to the HLS system. 2. L (luminance = brightness) is recalculated with $\text{NewLuminance} = 240 \cdot 3/4 + L/4$ 3. The color value is recalculated to the RGB system with the new brightness. <p>The color is <i>always</i> defined by the source with the highest priority!</p>
normal	The element is displayed in the color of the source and with the configured width.
Use alias	Active: Alias is used.
Alias	Opens the dialog (on page 39) for selecting a model.



Information

The source color and the priorities of the sources are defined in the project properties.

User-defined sources must have a higher ID than 9. IDs up to 9 are reserved for system sources.



Information

The calculation of the color of a line in the Runtime is done with the following priority list:

1. Automatic Line Coloring (highest priority, overrules all other settings)
2. Dynamic colors
3. Static colors


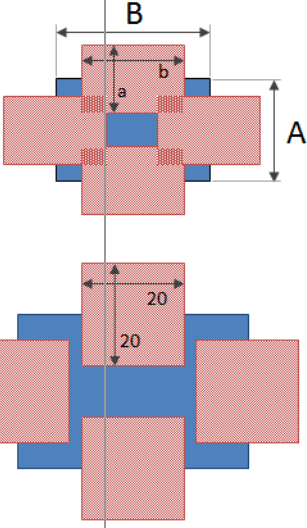






Example

In the following example Source 0 has the color blue and Source 1 has the color red. And Source 0 is the source with the highest priority.

 Source 0

 Source 1

This results in the following displays for the different options:

	Line / Polyline	Pipe
highest priority source		
two highest priority sources		
double width		
double brightness		

Connection points of lines

The connection of one line (line, polyline or tube) to another line is done with overlapping drawing in the screen at connection points. The connection points - either connection areas - are at the start and the end of each line and are around 3 pixels large.

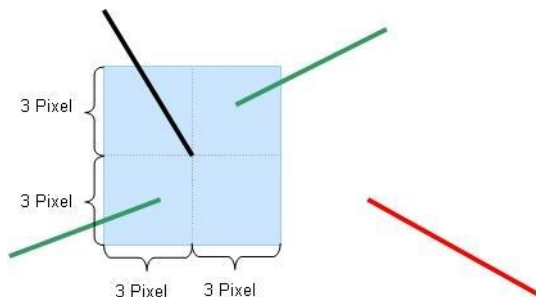


Example

*The start point of a line has the coordinates (start point x/start point y): 150/100 pixels.
This results in a connection area (x / y): 147 - 153 / 97 - 103 pixels.*

If the line start or end of this line and that of one or more other lines is within this area, the lines are automatically connected without any further engineering. A mere overlapping of the connection areas of the single lines is not sufficient!

In the following illustration the connection area is displayed graphically (the green lines are connected to the black one, the red line not).



Information

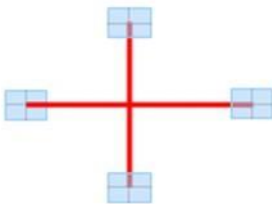
Any number of lines can be connected in a connection area.



Attention

If a line is outside the connection area (e.g. the red line in the illustration), no connection is established and there is no coloring of the line. So there will also be no coloring for further lines.

Line crossings can easily be realized, if the ends of the lines are not in the connection area.



Attention

Use ALC elements only in un-rotated state because:

The calculation for the topological model for the ALC in the Editor is based on the position of the elements in un-rotated state and without considering any dynamics.

3.1.3 Checking the project

Engineer the desired procedural elements and lines in one or more screens and save these screens. Then you can check via **Create all Runtime files** or **Create changed Runtime files** whether there are

any errors or conflicts in the screens. If error or conflicts should exist, corresponding error messages or warnings are displayed in the output window.



Information

Double click the corresponding line in the output window. The screen with the erroneous screen element will be opened automatically. If the erroneous screen element is part of a symbol, the corresponding symbol is automatically selected.

The following error message can be displayed.

- ▶ ALC: Screen '%s' - Two Link elements with different Link number are connected to line '%s' . (Double click opens the screen and selects the line.)
- ▶ ALC: Screen '%s' - More than two connection points are used at element '%s'. For each element only one input and one output may be used. (Double click opens the screen and selects the element.)

The following warnings can be displayed.

- ▶ ALC: Screen '%s' - Alias line '%s' is connected to a no-alias line. (Double click opens the screen and selects the line.)
- ▶ ALC: Screen '%s' - Alias element '%s' is connected to a no-alias line. (Double click opens the screen and selects the element)
- ▶ ALC: Screen '%s' - No-alias element '%s' is connected to an alias line. (Double click opens the screen and selects the element)
- ▶ ALC: Screen '%s' - Line '%s' is only connected on one side. (Double click opens the screen and selects the line.)
- ▶ ALC: Screen '%s' - Element '%s' is not connected. (Double click opens the screen and selects the element)
- ▶ ALC: Screen '%s' - Element '%s' is only connected on one side. (Double click opens the screen and selects the element)

In the error messages or warnings the corresponding elements are identified using the element reference. This reference also serves as the link key for ALC aliases.

3.2 Configuration

To configure ALC:

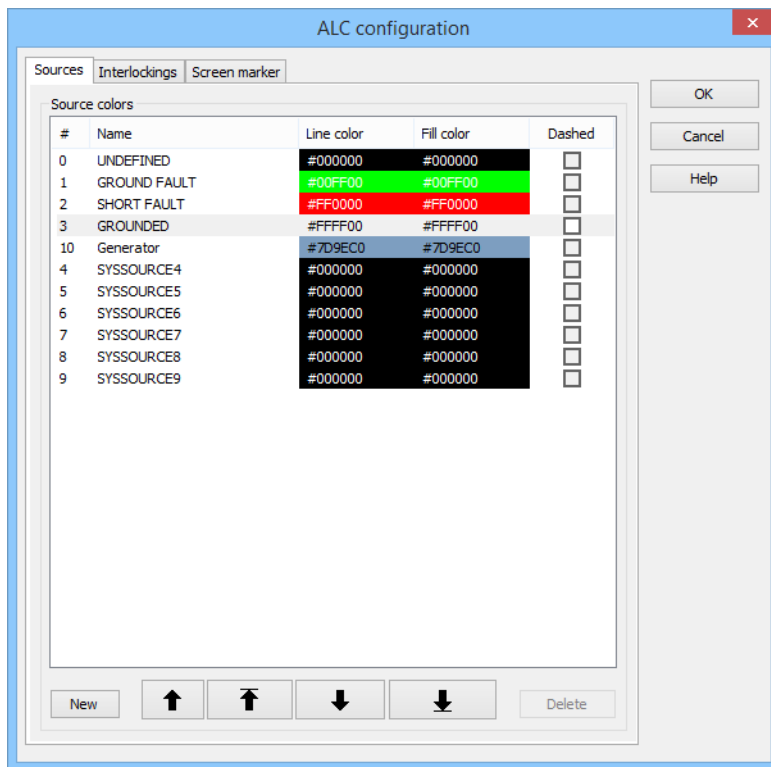
1. In project properties, select **ALC configuration** the property in the **Automatic Line Coloring** group
2. Click on the ... button
3. The dialog for configuration is opened

4. Configure the desired properties for:

- Sources (on page 28)
(note also the principles for Coloring for UNDEFINED (on page 31).)
- Interlockings (on page 32)
- Screen marker (on page 36)

3.2.1 Configuration of the sources

The sources, e.g. their names and colors (sequence and priority), are configured project-specifically within the project properties under 'ALC configuration'.



SOURCE COLORS

Parameters	Description
Number	Internal unique consecutive number, so that the source can be identified. This number is given by the system automatically and cannot be changed. Attention: IDs 0 to 9 are reserved for the system sources and must not be used user-specific.
Name	Logical name for the source (e.g.: 'water' or 'grounded'). This name is also used when selecting the source number for Combined elements. You can change the name by clicking it with the left mouse button. With this edit mode is switched on. The changes are accepted with Enter or by selecting another source. Note: The labels are not language switchable.
Line color	Line color of the respective source. This color is used for coloring lines, polylines and as the outside color of tubes.
Dashed	Check box for activation. If active, the line is drawn as dashed. Note: This check box is only active for GROUNDED . This check box is grayed out for all other sources.
New	Adds a new color.
Delete	Deletes the selected color.
Upwards (arrow symbol)	Moves selected source up one position.
Fully upwards (arrow symbol)	Moves selected source to the start of the list.
Downwards (arrow symbol)	Moves selected variable down one position.
Fully downwards (arrow symbol)	Moves selected source to the end of the list.

CLOSE DIALOG

Parameters	Description
OK	Applies all changes in all tabs and closes the dialog.
Cancel	Discards all changes in all tabs and closes the dialog.
Help	Opens online help.

The colors can be configured directly by entering the corresponding hexadecimal code or by using a color palette.

For direct input:

1. Click on the color description with the left mouse button
2. The field is switched to editing mode
3. Enter the code
4. Press the input key or select another source in or to accept the change

To select via a color palette:

1. highlight the desired line
2. click on the ... button behind the color
3. Online help is opened.
4. select the desired color

The hexadecimal code describes the RGB color value and consists of the following. #RRGGBB.

Element	Meaning
#	Identifier to indicate that a hexadecimal color code is used.
RR	2 digits are the red value of the color in hexadecimal system. 0-255 is 0-FF
GG	2 digits are the green value of the color in hexadecimal system. 0-255 is 0-FF
BB	2 digits are the blue value of the color in hexadecimal system. 0-255 is 0-FF



Information

The sequence in this list represents the priority of the sources, with the first element having the highest priority.

To change the priorities of the single sources, they can be moved upwards or downwards using the arrow buttons



Attention

Limitations when deleting the sources and resetting erroneous colorings:

Sources with ID between 0 and 9 are reserved for system sources. They can:

- ▶ Not be deleted:
- ▶ Not be reset as an erroneous color

Deleting sources

In order for sources to be able to be deleted, they must have an ID from 10. Only the source with the highest ID can be deleted.

Resetting erroneous colorings

In order for erroneous colorings to be able to be reset once the cause has been rectified, no system source colors can be used. A color for IDs from 10 must be selected.

Coloring mode for UNDEFINED

Coloring in the network can be implemented in two modes with the UNDEFINED status:

- ▶ Standard
- ▶ Input takes priority

This setting is made using the **Automatic Line Coloring/Mode for coloring** project property.

STANDARD

The graph search starts with a source and goes through the whole network, so that each closed switch (switch variable has the value 1) per direction is only gone through once, so no cycles occur. In doing so, each node visited (=line segment) is colored with the source color. The directly-related lines are marked as a node.

If the search finds a switch that has a switch variable with the following status, the UNDEFINED color is used for coloring from this point onwards:

- ▶ INVALID [values: any desired],
- ▶ is invalid [value: 3]
- ▶ is in intermediate position [value: 2])

The graph search is now continued in the same form. Each switch is gone through just once per direction with the UNDEFINED color. Therefore each switch can be gone through a maximum of four times per source:

1. with source number in forwards direction,
2. with source number in backwards direction,
3. with UNDEFINED in forwards direction,
4. with UNDEFINED in backwards direction,

INPUT TAKES PRIORITY

With the `Supply takes priority` setting, only lines that have a supply from at least one source but not clearly from any one source are colored as UNDEFINED. If a line is supplied with at least one source, it can no longer receive an UNDEFINED color from another source.

This search is a two-stage search:

- ▶ In the first stage, as with `Standard`, the source color is distributed in the network from each switched source, as long as the next switch is closed. The search is ended if the switch is open or invalid/undefined.
- ▶ In the second stage, the search is started at each invalid/undefined switch that receives a supply from one side and the UNDEFINED color is distributed to the unsupplied side. This search also considers the switches that are invalid/undefined as closed and thus distributes the UNDEFINED color in the network until it meets a clearly open switch. In addition, a search is ended if a line element is reached that is already supplied.

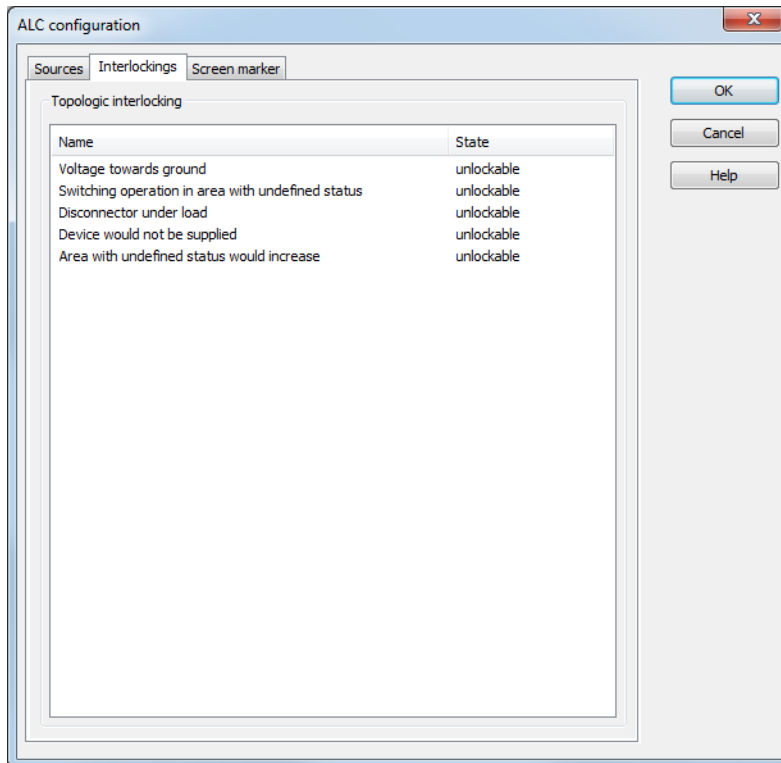
3.2.2 Configuration of topological interlockings

topological interlockings from the ALC for Command Processing can be configured here.



Information

*This dialog is only available when both the Energy Edition and the **Automatic Line Coloring** modules are licensed.*



The following conditions are available: The settings made here apply globally, for the whole Topological Model:

Parameters	Description
Voltage towards ground	Interlocking is active if a switch/disconnector is to be closed and a grounded potential is connected to its first connector and its other connector is connected or undefined.
Switching action in an area with an undefined status	Interlocking is active if a switch disconnector is to be closed and both of its connectors are 'undefined' or 'disturbed'.
Disconnecter under load	Interlocking is active if certain conditions have been met for switching on or off. Conditions: See "Disconnecter under load - interlocking conditions (on page 34)" section.
Device would not be supplied	Interlocking is active, when a consumer, which was supplied before, would be unsupplied after the switching action (by switch or disconnector).
Area with undefined status would increase	Interlocking is active if a switch disconnector is to be closed and one connector is 'undefined' or 'disturbed' and the other not.

If you click in the **Status** column in one of these interlockings, a drop-down list opens with three choices:

Parameters	Description
do not check	The selected condition is not considered in this project (topological model).
unlockable	The selected condition is considered in this project. If the condition applies, the user can unlock it with the command processing (Command Processing screen). This unlocking action is logged in the Chronological Event List.
not unlockable	The selected condition is considered in this project. The user cannot unlock it.

EXCEPTION TOPOLOGICAL INTERLOCKING

The topological interlocking is not carried out if:

- ▶ the variable of a switch has the status Revision
or
- ▶ the variable is manually corrected or set to **Alternate value** and with this is set to the same variable value as the initial value; in other words if the switch:
 - Is set to OFF and then it is manually corrected to OFF or replaced.
 - Is set to On and then it is manually corrected to ON or replaced.

Disconnecter under load - interlocking conditions

For the **disconnecter under load** topological interlocking, a disconnector can be switched if one of the following conditions is met:

WHEN BEING SWITCHED ON:

Before being switched:

- ▶ The neighboring line segment receives energy from the same source
- ▶ If the line segment does not receive any voltage, the other line segment is grounded
- ▶ A line segment is not live

WHEN BEING SWITCHED OFF:

After being switched:

- ▶ Neighboring line segments would not receive energy from the same source
- ▶ If one line segment does not receive any voltage, the other line segment is grounded
- ▶ A line segment would not be live



Information

Meaning of "not under load"

The status `not under load` means:

- ▶ Either:
All switches and disconnectors connected to the line segment are open.
- ▶ Or:
All switches and disconnectors connected to the line segment are closed but only connect to a segment that is also not under load.

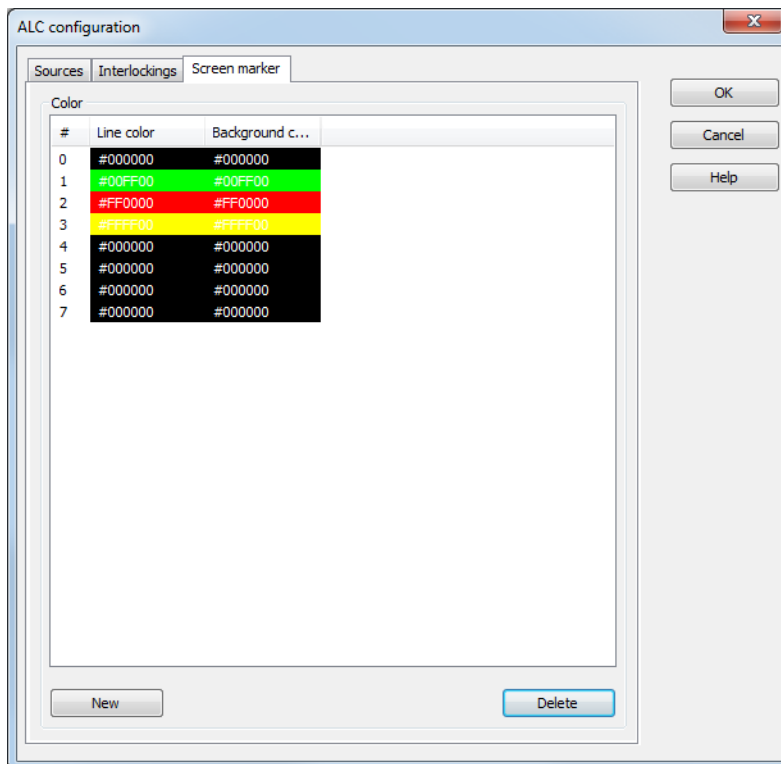
In addition, all of the following conditions must be met for the status of `not under load`:

- ▶ All sources and consuming devices connected to the line segment are switched off.
- ▶ No transformer may be connected to the line segment.
- ▶ It must not be a line that is only connected to this disconnector (open line).

▶

3.2.3 Configuration of the screen marker

Here you configure the color table for the color marker for the impedance-based error detection and calculation of load distribution (on page 54). See also: AddMarker



Parameters	Description
Number	Unique internal serial number for clear assignment. This number is given by the system automatically and cannot be changed.
Line color	Line color of the screen marker.
Fill color	Fill color of the screen marker.
New	Adds a new color.
Clear	Deletes the selected color.

The colors can be configured directly by entering the corresponding hexadecimal code or by using a color palette.

For direct input:

1. Click on the color description with the left mouse button
2. The field is switched to editing mode

3. Enter the code
4. Press the input key or select another source in or to accept the change

To select via a color palette:

1. highlight the desired line
2. click on the ... button behind the color
3. Online help is opened.
4. select the desired color

The hexadecimal code describes the RGB color value and consists of the following. #RRGGBB.

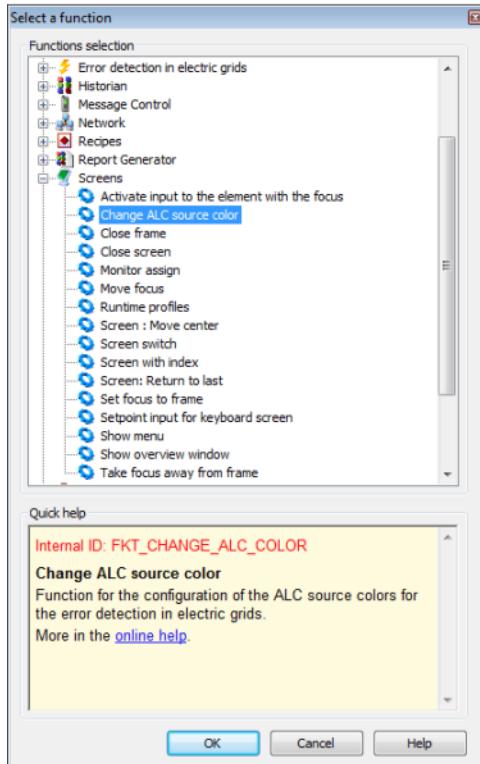
Element	Meaning
#	Identifier to indicate that a hexadecimal color code is used.
RR	2 digits are the red value of the color in hexadecimal system. 0-255 is 0-FF
GG	2 digits are the green value of the color in hexadecimal system. 0-255 is 0-FF
BB	2 digits are the blue value of the color in hexadecimal system. 0-255 is 0-FF

3.3 Change ALC source color

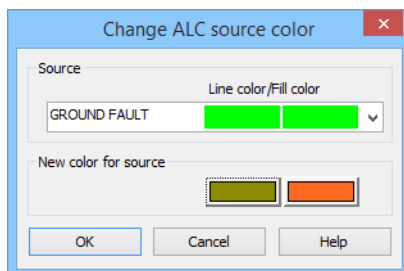
The foreground and background color of an ALC source can be temporarily changed for the coloring in Runtime using the **Change ALC source color** function. The change remains until Runtime is ended, reloaded or the function is executed again. To create the function:

- ▶ select New Function
- ▶ Navigate to the Screens node

- ▶ select **Change ALC source color**



- ▶ The dialog to define line colors and fill colors opens
- ▶ define the desired color



Property	Function
Source	Drop-down list to select the source and display the colors currently assigned. These colors cannot be changed here.
New color for source	Click on the color and a dialog opens to select a color.

3.4 Detail screens

To display individual screens, a partial area can be taken from the topological network and displayed individually by means of alias. The screen elements in the detail screen are not included in the topological model, but do however get their ALC colors from the model. They relate to an alias of the screen elements in the overall screen.



Attention

Aliases are only valid within a project.

This means that for symbols that contain links to aliases:

*If the symbol is added to the **general symbol library** or the **library in the global project** and edited there, all ALC alias information is lost without notice!*

CREATE ALIAS

Aliases can be created for the elements:

- ▶ Line
- ▶ Polyline
- ▶ Pipe
- ▶ Combined element



Attention

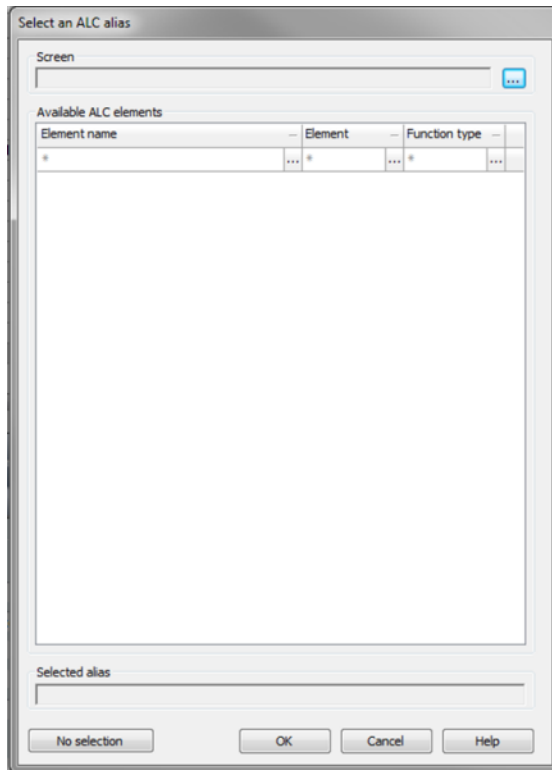
An ALC alias cannot be created if a period (.) is contained in the name of the selected screen.

Solution: Replace the period in the screen name with a different character, such as an underscore for example (_).

To create a source element as an alias:

- ▶ Activate it in the element's properties **Use alias**.
To do this, ALC must be licensed and the **Color from ALC** property active.
- ▶ Click on the ... button in the **Alias** property

- The dialog to select the element opens.



Parameters	Description
Screen	Click the ... button and a dialog opens to select a screen.
Available ALC elements	<p>Shows the elements that belong to a screen with the element name, type of element and function type. Clicking on an element selects an alias.</p> <p>Filter</p> <p>The elements can be sorted according to all columns. When setting a filter, the options offered from all other filters are reduced to values that can be sensibly combined.</p> <ul style="list-style-type: none"> ▶ Name: Input of a standard search term with wild cards (*). The last 12 search terms are offered in the list until the Editor is ended. ▶ Element: Select from drop-down list. ▶ Function type: Select from drop-down list. <p>Clicking on ... opens saved search or drop-down list.</p> <p>If a filter is active, clicking on the X deletes the filter.</p>
Selected alias	Shows the selected element in the field of Available ALC elements .
No selection	Removes selected element.
OK	Saves selection and closes dialog.
Cancel	Discards changes and closes dialog.
Help	Opens online help.



Information

When selecting an element for a new alias, only elements and screens from the same project that the alias was defined in can be selected. Elements from subprojects or parallel projects are not available.

REPLACING ALIAS NAMES

Aliases can be changed when switching screens with Replace link. A detail screen can therefore be displayed with the data from different equipment parts, for instance lines or partial networks. Alias names are replaced along the lines of variables and functions. It is also possible to replace in elements that are used in symbols. The same dialog as is opened for the target as the **Alias** property.

Note: Substitution using index variables is not possible.

3.5 Fault locating in electric grids

Error detection marks grid parts that are subject to ground faults or short circuits by means of special colors in ALC. Sources for error detection are what are called ground fault or short circuit reporters that are assigned to a circuit breaker. Ground fault and short circuit reporters are always at the output of a circuit breaker element. Error messages are fixed in the screen and must be reset manually.



Information

This function is only available when both the "Energy Edition" and the "Automatic Line Coloring" modules are licensed.

ERROR DETECTION

Error detection runs locally. Each client in the network has its own independent model and can therefore search for ground faults and short circuits in different parts of the network.

Error detection in the electrical network is divided into:

- ▶ Search for ground fault (on page 43)
- ▶ Short circuit search (on page 50)

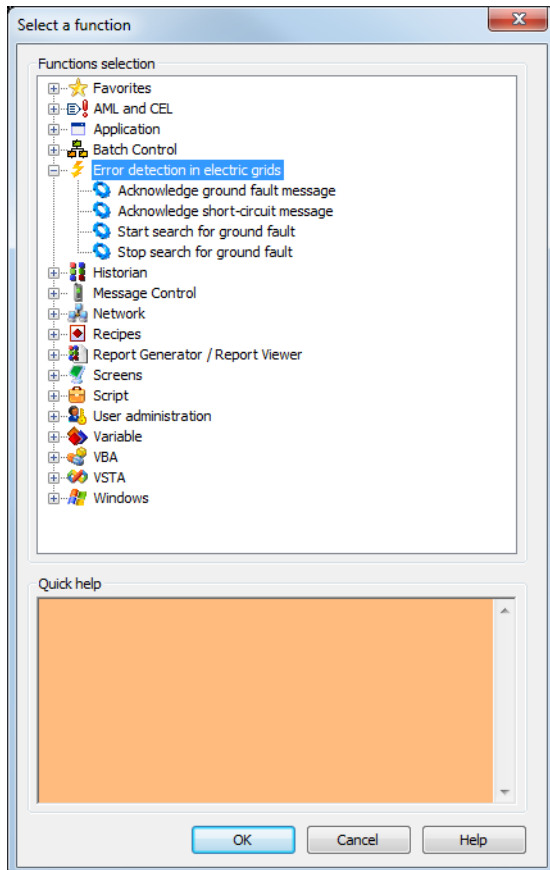
To configure error detection

- ▶ You require a license for ALC and zenon Energy Edition
- ▶ configure the appropriate screens
- ▶ Configure (on page 9) ALC to the corresponding combined elements (switch, transformer, disconnector, slider)
- ▶ configure (on page 20) the lines so that they are colored by ALC

Special functions are available in Runtime for error detection:

- ▶ Start search for ground fault (on page 47)
- ▶ acknowledge (on page 48) ground fault message (on page 48)
- ▶ Stop search for ground fault (on page 49)

- Acknowledge short-circuit message (on page 53)



COLORINGS

Errors can be shown by a special coloring for the lines in ALC. In Runtime, the color assigned by ALC changes automatically as soon as the status of the line changes. The colorings configured can be changed in Runtime via the change ALC source color (on page 37) function.

Messages are processed in the order in which they arrive. In the event of conflicts

- The colors for displaying errors take priority
- short circuit messages have priority over ground fault messages

3.5.1 Search for ground fault

The search for a ground fault serves to highlight the network parts that may have a ground fault by coloring these. The color is taken from the engineering of ALC source colors (on page 27) for the GROUND FAULT source.

The network parts that may have a ground fault are derived from the ground fault reports from ground fault detection devices (ground indicators, protective device that records ground faults). The following is applicable for ground faults:

- ▶ Each device can have one to three ground fault reports.
- ▶ Ground fault reports are handled either by permanent message processing or by wiper message processing.
- ▶ For directional ground fault detection devices, the direction can be lagging or leading in relation to triggering.
 - Leading: First comes the report, then the wiper bit.
 - Lagging: First comes the wiper bit, then the report.



Information

A network component that may have a ground fault is then no longer considered to have a ground fault if this has been successfully connected.

ENGINEERING

To configure a search for a ground fault:

1. assign the combined element that represents the switching element to the **Function type** switch (on page 45)
2. Define the mode of search for ground fault (on page 44), ground fault display (on page 46) and ground fault detection triggering (on page 46)
3. Create the functions for start search for ground fault (on page 47), acknowledge ground fault report (on page 48) and end search for ground fault (on page 49)



Information

In order to also be able to limit ground faults in mixed networks, only one area with ground faults is searched per path, starting with a source.

Mode of the search for ground faults

The short circuit search can either:

- ▶ color the network part potentially subject to a short circuit
or
- ▶ the whole network where the short circuit is located

The coloring mode is defined via the **Mode of the search for ground faults** property.

To configure the property:

- ▶ navigate to the **Automatic Line Coloring** node in properties
- ▶ select the desired mode in the **Mode of the search for ground faults** property drop-down list
 - Color grid part: colors only the grid parts that are potentially subject to a short circuit
 - Color whole grid: colors in the whole linked grid where the short circuit is located

This setting can be changed in Runtime via the zenon API object model. In doing so, the short circuit search is recalculated once again.

Ground fault detection type

The direction and type of message processing for the combined element are determined by means of the **Type** setting. For project configuration:

1. navigate to the **Automatic Line Coloring** node in the combined element properties
2. open the **Ground fault recognition** node
3. Select the desired type with direction and type of message processing in the **Type** property
 - Direction:
indicates if the raising edge of trip alarm or if the raising edge of a direction comes before
 - leading: The current direction status is used for the raising edge of the trip alarm
 - lagging: after a raising edge of the trip alarm, the first raising edge of a direction is waited on; if this does not occur within 2 seconds, the earth fault device is considered non-directional
 - Information processing:
states which information can be processed
 - none: normal switch; information is not processed
 - Permanent message processing: Newly received messages are considered a new ground fault trip
 - wiper message processing: Messages that are received during a current Search (on page 47) are suppressed

Note: The distinction between permanent message processing and wiper message processing is only how the message is processed, not its type. Wiper message processing thus does not need to relate to a wiper bit.



Attention

To suppress intermittent ground faults, ground fault messages that occur in intervals of less than 2s are ignored.

Ground fault display

The variable linked at **Display** is an output variable for error detection and displays the recorded status of the ground fault identification device. This is necessary because all messages remain saved internally until they are acknowledged, i.e. they do not necessarily conform to the current status of the message variables.

Each time a recording is made, a set value is sent to this variable. In doing so, the values are as follows:

Value	Meaning
0	no ground fault
1	ground fault forwards
2	Ground fault backwards
3	non-directional ground fault
4	Error status - > both directions have activated



Information

To reduce problems in network operation, the variable linked here should be a linked variable.

Earth fault triggering

The alarm to report an earth fault is defined by the **Triggering** variable. It can contain information on the presence of an earth fault and the direction of the earth fault from the point of view of the earth fault recognition device. In doing so, a distinction is made between:

- ▶ non-directional earth fault alarms
- ▶ Directional earth fault alarms with a trip alarm
- ▶ Directional earth fault alarms with a trip alarm

To configure the variable for the **Triggering**:

1. navigate to the **Automatic Line Coloring** node in the combined element properties
2. open the **Ground fault recognition** node
 - a) for non-directional earth fault alarms

Click on the ... button in the **Triggering** property

select the variable you wish to import in the dialog that opens

The properties for the direction remain empty

- b) for directional earth fault alarms with a trip alarm
link the variable with **Triggering** and add the appropriate direction:
Forwards: link a variable to the **Forwards** property
Backwards: link a variable to the **Backwards** property
- c) for directional earth fault alarms without a trip alarm
Link the variable with the corresponding direction:
Forwards: link a variable to the **Forwards** property
Backwards: link a variable to the **Backwards** property
The **Triggering** property remains empty

Note: If you address a directional identification device with **Forwards** in both directions, this is then considered erroneous and ignored.

Start search for ground fault

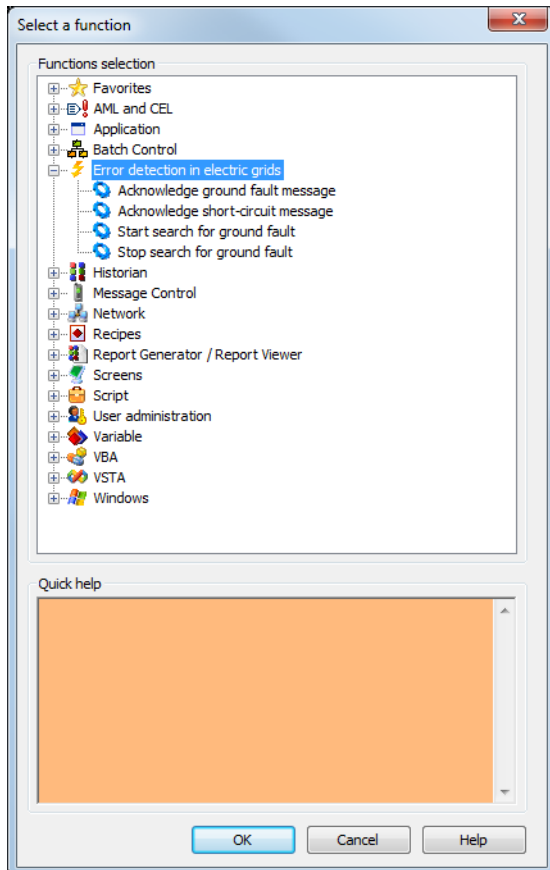
The function **Start search for ground fault** serves to localize a ground fault and has two effects in Runtime:

1. Fault reports from all ground fault identification devices that were configured with wiper message processing are ignored.
2. The search algorithm is changed: Switch actions can only reduce the area subject to a ground fault further. Newly received messages do not therefore increase the area potentially subject to a ground fault.

To configure the **Start search for ground fault** function:

- create a new function
- navigate to the error detection node in the electrical network

- Select the **Start search for ground fault** function



- link the function to a button

Acknowledge ground fault message

With the **Acknowledge ground fault message** function, an internally recorded ground fault from a ground fault indication device can be acknowledged. In doing so, the internally-latched ground fault status is reset if the status is still pending, or highlighted as acknowledged. A recorded ground fault message is only deleted internally if this has been acknowledged and is no longer pending.

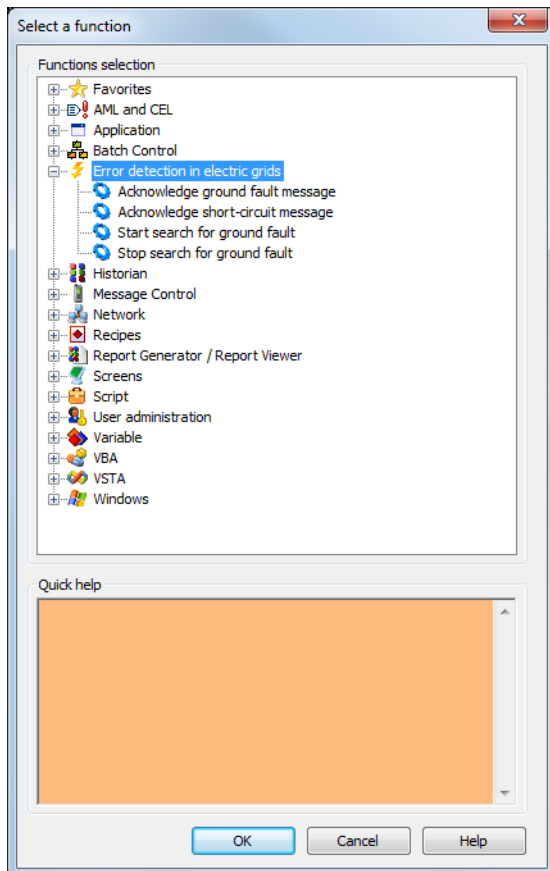
Rules when acknowledging:

- If a variable that corresponds to a triggering or direction variable of a ground fault recognition device is linked, this special ground fault message is acknowledged.
- If no variable has been linked, all ground fault messages are acknowledged.
- Acknowledgment can also take place via the zenon API object model.

To configure the **Acknowledge ground fault message** function:

- create a new function

- ▶ navigate to the error detection node in the electrical network
- ▶ Select the **Acknowledge ground fault message** function



- ▶ the dialog to select a variable opens
- ▶ link the desired variable to the function
- ▶ link the function to a button

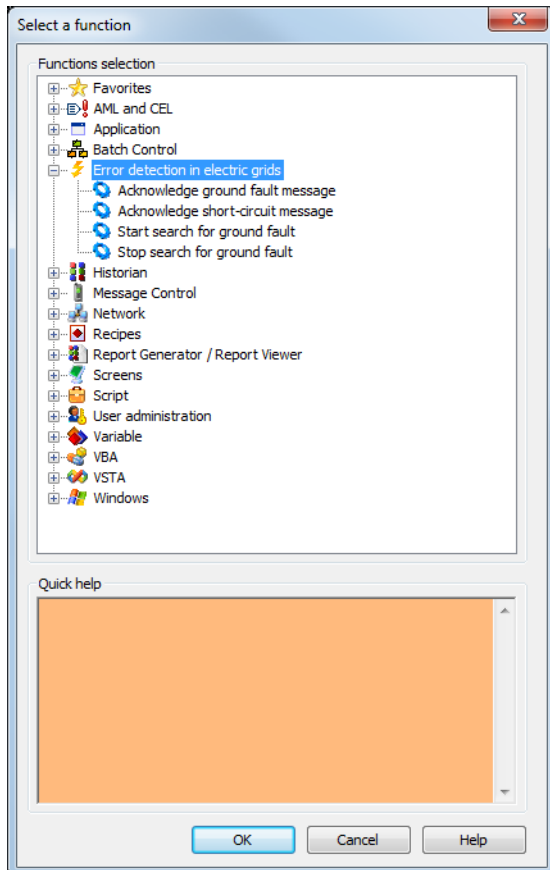
Stop search for ground fault

You end the ground fault search with the **Stop search for ground fault** function in Runtime.

To configure the function:

- ▶ create a new function
- ▶ navigate to the error detection node in the electrical network

- Select the **Stop search for ground fault** function



- link the function to a button

3.5.2 Short circuit search

The short circuit search serves to highlight the network parts that potentially have a short circuit by coloring these. The color is taken from the configuration of ALC source colors for the `SHORT FAULT` source.

The network parts that are potentially subject to short circuits are deduced from short circuit reports. A short circuit identification device (short circuit indicator, protective device) can have one to three short circuit messages. For directional short circuit indication devices, the direction can be lagging or leading in relation to triggering. A network component that potentially has a short circuit is then no longer considered to have a ground fault if this has been successfully connected.

ENGINEERING

To configure the short circuit search:

1. assign the combined element that represents the switching element to the **Function type** switch (on page 51)
2. Define ground fault display (on page 51) and triggering of ground fault detection (on page 52)
3. Set up the function for acknowledgment of ground fault message (on page 53)

Ground fault detection

The direction and type of message processing for the combined element are determined by means of the **Type** setting. For project configuration:

1. navigate to the **Automatic Line Coloring** node in the combined element properties
2. open the **Short-circuit detection** node
3. Select the desired type in the **Type** property
 - **Direction:**
indicates if the raising edge of trip alarm or if the raising edge of a direction comes before
 - **Leading:**
The current direction status is used for the raising edge of the trip alarm
 - **lagging:**
after a raising edge of the trip alarm, the first raising edge of a direction is waited on; if this does not occur within 2 seconds, the short circuit device is considered non-directional
 - **Information processing:**
states which information can be processed
 - **none:**
normal switch; information is not processed
 - **Permanent message processing:**
Newly received messages are considered a new ground fault trip

Ground fault display

The variable linked for **Display** is an output variable for error detection and displays the recorded status of the ground fault detection device. This is necessary because all messages remain saved internally until they are acknowledged, i.e. they do not necessarily conform to the current status of the message variables.

Each time a recording is made, a set value is sent to this variable. In doing so, the values are as follows:

Value	Meaning
0	No short circuit
1	Short circuit forwards
2	Short circuit backwards
3	Non-directional short circuit

Ground fault detection triggering

The variable for the message from the short circuit identification device is defined by the **Triggering** variable. It can contain information on the presence of a short circuit and the direction of the short circuit from the point of view of the ground fault recognition device. In doing so, a distinction is made between:

- ▶ non-directional short circuit reporters
- ▶ directional short circuit reporters with a trip alarm
- ▶ directional short circuit alarms with a trip alarm

To configure the variables for:

1. navigate to the **Automatic Line Coloring** node in the combined element properties
2. open the **Short-circuit detection** node
 - a) for non-directional short circuit detection devices

Click on the ... button in the **Triggering** property

select the variable you wish to import in the dialog that opens

The properties for the direction remain empty
 - b) for directional short circuit detection devices with a trip alarm

link the variable with **Triggering** and add the appropriate direction:

Forwards: link a variable to the **Forwards** property

Backwards: link a variable to the **Backwards** property
 - c) for directional short circuit detection devices without a trip alarm

Link the variable with the corresponding direction:

Forwards: link a variable to the **Forwards** property

Backwards: link a variable to the **Backwards** property

The **Triggering** property remains empty

Acknowledge short-circuit message

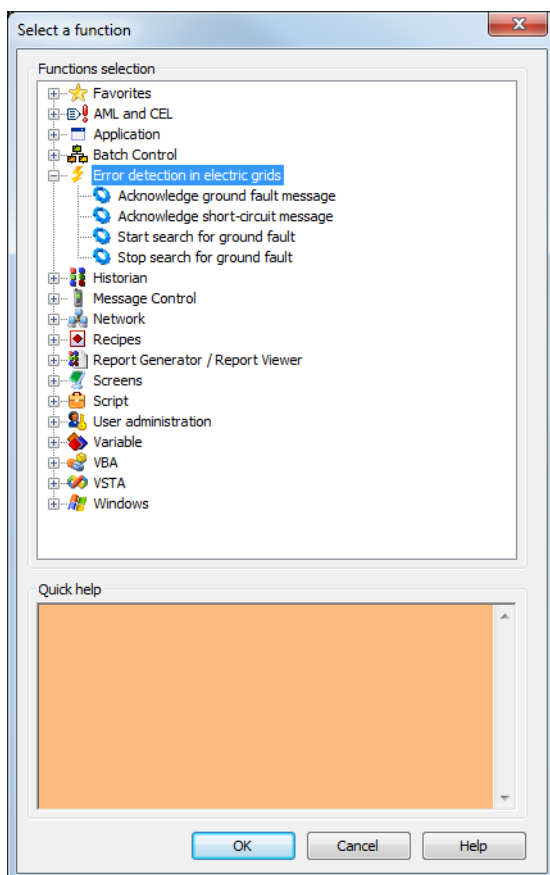
With the **Acknowledge short-circuit message** function, an internally recorded short circuit from a short circuit indication device can be acknowledged. In doing so, the internally-latched ground fault status is reset if the status is still pending, or highlighted as acknowledged. A recorded short circuit message is only deleted internally if this has been acknowledged and is no longer pending.

Rules when acknowledging:

- ▶ If a variable that corresponds to a triggering or direction variable of a short circuit recognition device is linked, this special short circuit message is acknowledged.
- ▶ If no variable has been linked, all short circuit messages are acknowledged.
- ▶ Acknowledgment can also take place via the zenon API object model.

TO CONFIGURE THE ACKNOWLEDGE SHORT-CIRCUIT MESSAGE FUNCTION:

- ▶ create a new function
- ▶ navigate to the error detection node in the electrical network
- ▶ Select the **Acknowledge short-circuit message** function



- ▶ select the variable you wish to import in the dialog that opens
- ▶ link the function to a button

3.6 Impedance-based error detection and calculation of load distribution

Impedance based error detection and calculation of load distribution expands ALC. Whereas ALC identifies nodes and beams, this model also detects lines and their parameters. The model is not used internally in zenon, but makes properties and methods for external evaluation available.

PROPERTIES FOR ALC AND THE EXTENDED TOPOLOGICAL MODEL

The ALC elements combined element and line (line, polyline, tube) have special properties for error detection for protection and to calculate the load distribution. These properties are not evaluated in zenon, but are available via the zenon API algorithms to be created by users.

The simple topological model for the coloring was supplemented by an expanded topological model that includes all lines as separate beams. The extended topological model is stored as **ALC.xml** and can be read by external applications this way. **ALC.xml** contains two sections:

- ▶ **GraphElements**: contains the extended topological model without aliases
- ▶ **GraphAliases**: contains only the aliases

Each object has a unique ID, via which it is referenced in the file. The attributes correspond to a subset of the zenon screen elements that have created the elements.

GRAPHELEMENT

ID	Description
Picture	Screen name
ElementID	Screen element ID
ElementRef	Screen element reference
Type	Screen element -type (see "element")
SourceID	Source number
ReverseSourceID	Source name in reverse direction
Variable	Status variable
VarProtReact	Reactance variable
MaxIType	Type of maximum current
MaxIVal	Maximum current constant value
VarMaxI	Maximum current variable
VarCurl	Instantaneous current variable
VarCalcl	Calculated current variable
VarCurP	Instantaneous power variable
LoadType	Type of load
LoadVal	Load constant value
VarLoad	Load variable
React	Reactance
Resist	Resistance
Length	Line length
Node1IDs	List of all element IDs connected with Node1
Node2IDs	List of all element IDs connected with Node2

GRAPHALIAS

ID	Description
Picture	Screen name
ElementID	Screen element ID
ElementRef	Screen element reference
Type	Screen element -type (see "element")

OrigElemRef	Screen element - reference to the original screen element
OrigGraphElemID	ID of the original elements in "GraphElements"

API

In the object model of the zenon API, the objects **ALCGraphElement** and **ALCGraphAlias** are available for the model. These contain the same information as the XML file. These objects can be accessed in the ALC engine via:

- ▶ **GraphElemCount()**
- ▶ **GraphAliasCount()**
- ▶ **GraphElemItem()**
- ▶ **GraphAliasItem()**

USER-SPECIFIC TOPOLOGICAL INTERLOCKINGS

If a topological interlocking is checked, the following event is called up at the ALC engine:

- ▶ **void CheckInterlocking(IALCEdge* pALCEdge, long nNewState, tpLockResult* LockResult, BSTR* bsText, VARIANT_BOOL* bUnlockable);**

The switch/disconnector to be switched and the new status is transferred. The event can fill **LockResult**, **bUnlockable** and **bsText** in order to display a violated interlocking condition. If the event handler returns **tpBusy** in **LockResult**, the event handler is queried until it no longer provides **tpBusy**, however for a maximum of 10 seconds. The interlocking is active after 10 seconds. The interlocking text and unlockability are reported back in **bsText** and **bUnlockable**.

SCREEN MARKER

Marker elements can be inserted into screen s via the zenon API.

These are added or deleted via the API functions in **DynPictures**:

- ▶ **BSTR AddMarker(BSTR bsScreenName, long nElementID, short nPosition, short nLineColorIndex, short nFillColorIndex);**
- ▶ **VARIANT_BOOL DelMarker(BSTR bsID);**

The GUID of the marker, which is supplied by **AddMarker()**, identifies the marker uniquely and serves as both the element name (with the prefix "**\$MARKER_**") as well as the key for deletion via **DelMarker()**. The markers inserted via API are saved in the project according to the screen. Attention: Saving is not remanant, i.e. only until Runtime is restarted.

The markers set there are displayed regardless of the monitor on which the screen is opened. The markers are treated internally as normally operable screen elements. Mouse events are called up for this.

The appearance of the markers is set using the project settings in the **Automatic Line Coloring** area of the project configuration:

- ▶ **Display type of the screen marker:** Triangle, circle, square, cross
- ▶ **Screen marker size:** Size in pixels:
- ▶ **Line width of the screen marker:** Width in pixels
- ▶ **Marker color:** is defined via the index in the marker color table (on page 36), that is located in the properties of the screen elements in the **Automatic Line Coloring** group

4. Command Processing

Command processing serves primarily for the secured switching of variables in energy technology. 'Secured' means that there is a check whether the switching operation is allowed, according to the configured interlocking condition and the dynamically updated topology (current physical state of the topological network). The configuration of the topology and the topological commands is done via the ALC (Automatic Line Coloring) (on page 7) module.



License information

Included in the license for Energy Edition.

Command groups always contain a set of defined actions, which are usually adjusted to a specific data point (a specific device) . For example, different command groups can be defined individually and centrally for different topological elements (switch / disconnecter etc.) .

A data point for the command processing always consists of 2 physical variables: a response variable and a command variable.

The response variable is defined centrally for the whole command group. It represents the status of the topological element. For example whether the switch is open or closed. A defined command variable is assigned to every action inside a command group. The driver uses this variable to write commands to the controller.

Depending on the action to be executed, these commands are executed on one of the two variables.



Example

'Switching command on'

Sends the command/the new value to the command variable. The success of the triggered action can be checked by means of the response variable.

Status input off'

Resets all configured status bits of the response variable. The command variable is not affected by this action (note: command variable is the response variable).

NAME REPLACEMENT

To simplify or to generalize the definition of the variables, the variable references (for command variables, response variables and condition variables) can be defined using a name replacement. In doing so, wildcards '*' can be used. Wild cards are only allowed as prefix or suffix, i.e. *xxx or xxx*

As a result of this flexible definition, generally-valid procedures can be defined, which are then applicable for several data points. The number of command groups that must be defined is thus reduced considerably.



Example

- ▶ Definition of the command variables '*_BE'
- ▶ Definition of the response variables '*_RM'

In Runtime, the Command Processing automatically adds the name of the response variable, which is shown/selected in the process screen, to the name of the command variable. The names of both variables differ only in their endings.

Two-step command processing operations are usually performed via a context menu and the screen type „Command Processing“. Specific control elements are available for this screen type. They enable an individual optical and functional design of the command processing. This way, individual actions, for example, can be assigned to **action buttons** directly. After this, these actions can be selected by the user directly. This screen type also includes the necessary requirements in order to carry out functions such as unlocking, two-step execution, two-hand operation, locking etc. Such a screen can be called up by means of a function at the screen element directly (instead of the set value dialog), or via a context menu.

From the command groups and the current status of the topological model (automatic line coloring - interlockings) it is evident in runtime whether an action (switching operation) is permitted at a certain time

- ▶ (no interlocking condition applicable),
- ▶ not permitted (there is a non-unlockable condition applicable) or
- ▶ can be executed after unlocking (unlockable condition applicable).

In the zenon network, there is synchronization of actions from the command processing, which concern a certain response variable - by activating the `NET_SEL` status bits. The parallel execution of different users on the same object (same variables) is thus precluded. Parallel execution on different response variables is supported.

PROJECT MANAGER CONTEXT MENU

Menu item	Action
New command group	Creates a new command group.
Export XML all...	Exports all entries as an XML file.
Import XML...	Imports measuring units from an XML file.
Editor profile	Opens the drop-down list for selecting a Editor profile.
Help	Opens online help.



Information

Command groups can be exported, imported and copied and pasted using the clipboard. The same applies for actions and their interlocking conditions, even different command groups.



Attention

All of the following functions are only available if zenon Energy Edition is licensed.

4.1 Command processing detail view toolbar and context menu



COMMAND PROCESSING AND COMMAND GROUP CONTEXT MENU

Menu item	Action
New command group	Creates a new command group.
Export XML all...	Exports all entries as an XML file.
Export selected XML..	Exports selected entries as an XML file.
Import XML...	Imports from an XML file.
Copy	Copies the selected command group to the clipboard.
Paste	Pastes command groups from the clipboard.
Delete	Deletes the selected command group after requesting confirmation.
Rename	Enables renaming of a command group.
Properties	Opens the properties window for the selected command group.
Help	Opens online help.

CONTEXT MENU GROUP ACTIONS

Menu item	Action
Command new	Creates a new command and opens the properties.
New auto/remote command	Creates a new auto/remote command and opens the properties.
New forced command	Creates a new mandatory command and opens the properties.
New set value input	Creates a new set value input and opens the properties.
New status input	Creates a new status input and opens the properties.
New replace	Creates a new replace action and opens the properties.
New revision	Creates a new revision and opens the properties.
New manual correction	Creates a new manual correction action and opens the properties.
New block	Creates a new block action and opens the properties.
New release	Creates a new manual correction and opens the properties.
Check response value	Creates a new check response variable action. Note: This action checks the status of a variable without writing a value.
New lock	Creates a new lock and opens the properties.

Paste	Pastes action from the clipboard.
Help	Opens online help.

CONTEXT MENU INDIVIDUAL ACTION

Menu item	Action
New interlocking condition	Creates a new interlocking condition. Note: Grayed out for mandatory command command processing actions.
Copy	Copies the selected action to the clipboard.
Paste	Pastes action from the clipboard.
Delete	Deletes the selected action after requesting confirmation.
Help	Opens online help.

CONTEXT MENU CONDITION

Menu item	Action
Remove interlocking condition	Deletes selected condition.
Copy	Copies the selected condition.
Paste	Pastes the condition from the clipboard.
Properties	Opens the property window for the selected element.
Help	Opens online help.

CONTEXT MENU GROUP VARIABLES

Menu item	Action
Add variable...	Opens the dialog for selecting a variable.
Paste	Pastes variable from the clipboard.
Help	Opens online help.

CONTEXT MENU INDIVIDUAL VARIABLE

Menu item	Action
Remove variable	Deletes the selected variable from the group after requesting confirmation.
Copy	Copies selected variables to the clipboard.
Paste	Pastes variable from the clipboard.
Properties	Opens the property window for the selected element.
Help	Opens online help.

4.2 Engineering in the Editor



Information

*For tests too, use a driver that supports the evaluation of the COT (Cause of Transmission - Cause of transmission) in full, for example the IEC 60870-5-101_104 driver. COT evaluation is an enhanced functionality to monitor communication during a command using the **Runtime monitoring** setting. The status bits COT_x of the command variables can also be evaluated in the reaction matrices **multi-numerical** and **multi-binary**.*

4.2.1 Creating a screen of the type Command Processing

The creation of the command processing screen in the editor is done by the definition of a new screen of the type **Command Processing**. (You will find more information on the pre-defined screen types in the manual Screens/Pre-defined screen types'.)

The screen 'Command Processing' is used for user interaction via command during the runtime (one and two-step command) It allows the user to perform all activities that are necessary for command execution. This can be, for example, the unlocking of an active command processing group or the confirmation of the execution of a two-step command.

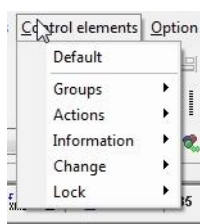


Information

When using the one-step command processing, you can also use a context menu or a standard function. The screen type command processing is then not required in the project.

You can use specific control elements (on page 128) for this screen type, which allow all user actions necessary for command processing and which visualize current information about the status of the action to be executed. (e.g. display of the switching direction)

After the screen is opened an empty screen is displayed. You add the default control elements via menu **Control elements/Add template**.



Variable substitution

For command processing - in addition to variable substitution of zenon - substitution rules can be configured per command group or command action.

When configuring a project in the zenon Editor, you can find the **Replace in screen** property for each command group or command action. This property is in the **Command Processing screen** property group.

REQUIREMENTS FOR SUBSTITUTION

Requirements for use are:

- ▶ The response variables and action variables were configured in the command processing with the * (star) placeholder.
 Example: *_RV
 Response variable: **Variable name of response**
 Command variable: **Action variable**
- ▶ A command processing screen is assigned in the **Screen** in the respective command group or command action.

Variables are substituted according to the following rule in the command processing screen:

- ▶ The text from the property is substituted in the variable name that is shown in the command field.
- ▶ It is substituted with a text which command found in the name of a response variable or action variable in place of the placeholder *.

Several texts to be substituted are configured separately with a semicolon (;). These phrases are substituted from left to right when calling up a screen in Runtime. The following phrases are ignored as soon as a text for replacement is applied.

If, when calling up the command processing screen in Runtime, there is no variable name with the configured text, there is also nothing substituted.

SCREEN SWITCHING AND COMMAND PROCESSING SUBSTITUTION RULES

Substitution via the screen switching function can be combined with the substitution of command processing. The following rules apply for substitution:

- ▶ If the screen is called up with a **Screen switch** function, the substitution configured in the function is used in Runtime.
- ▶ If the screen is called up using the **Command Sequencer** module or the menu, Runtime gets the screen and the substitution from the project configuration in the respective command action. If there is no substitution configured in the command action, Runtime gets the screen and the substitution from the command group. If there is also no substitution configured in the command group, there is no replacement.
- ▶ When clicking on a dynamic element that has `new set value input` configured, Runtime gets the screen and the substitution from the **setpoint input** command action.



Information

You can get further information on substitution via the screen switching function in the command processing chapter in the functions and scripts manual.

EXAMPLE

The following are configured:

- ▶ Value of the response variable: `abc_RV`.
- ▶ Configured response variable for command group: `*_RV`
- ▶ Existing variables in the screen: `xyz_lock`, `xy_Switch`

Result - scenario 1

- ▶ Configured **replacement in the screen**: `xyz;xy`
- ▶ Existing variables in the project: `abc_lock` and `abc_Switch`.
- ▶ **Result**: Display of the variables in the screen: `abc_lock` and `abc_Switch`.

Result - scenario 2

- ▶ Configured **replacement in the screen**: `xy;xyz`
- ▶ Existing variables in the project: `abcz_lock` and `abc_Switch`.
- ▶ **Result**: Display of the variables in the screen for `abcz_lock` and `abc_Switch`.

Result - scenario 3

- ▶ Configured **replacement in the screen**: `xy;xyz`
- ▶ Existing variables in the project: `abc_lock` and `abc_Switch`.
- ▶ **Result**: Display of the variables in the screen: `abc_lock` and `abc_Switch`.
Because `abcz_lock` is not present.

4.2.2 Variables of the command group

Command groups use both the variables of the switching actions (the response variable and command variable) and the variables of the command conditions.

In order for variables to be able to be used in the command processing module, a command group must also be assigned for the respective variable (variable of the command condition). This assignment is made in the variables node => for the variable => in the **Write set value** properties group => in the drop-down menu of the **Command Group** property.

Ensure that this assignment is configured for both response variables and command variables.

If a response variable or a command variable is linked to a screen element, the set value can only be set using command input.

The values of such variables can be written directly:

- ▶ via the **RGM**
- ▶ via **API**
- ▶ In zenon Logic with the **Externally visible** property activated.

The command input is ignored in the process.



Information

*If a variable is linked to a command group, it is not possible to describe the variable with the zenon **Write set value** function.*

*Exception: If a **write set value** (on page 80) command with **switching direction set value** has been created, the zenon function calls up this action in the background without the **command input** screen being called up. This means that the **command conditions** (on page 99) are checked. An active interlocking condition prevents the writing of a set value. During the execution of an action, the **NET_SEL** status bit is not set and the **Select Before Operate** variable property is ignored.*

*This is also applicable for the value entry of a variable that is linked to a dynamic element if **Element** was selected for the **Write set value via property**.*

GENERAL EXAMPLE

Assuming that that command group "DPI one stage" was configured with the name of the response variable *_RV and the switching actions in this group with the name of the action variable *_CO.

In the SCADA project, variables with the name `ied9_100_RV` (position of the switch) and `ied9_100_CO` (command for switch) are configured.

Then link the two variables `ied9_100_RV` and `ied9_100_CO` in the command group property to the "DPI one stage command group". The respective wild card * is replaced with "ied9_100 in the Runtime. This way, other variables, such as `ied9_101_RV` and `ied9_101_CO` (etc.) can be linked to this command group.



Information

As a result of the different use of limit values / reaction matrices for the command variable / return variable, individual switching directions can be displayed for the actions. Always depends on the variables on which the action to be executed is carried out.

Limit values and reaction matrices for switching direction texts

In the first step of the two-step command processing, user-definable switching direction texts are displayed (e.g. in the context menu) for **Command** actions. These texts can be defined via the limit values or via the states of the reaction matrices.

You therefore also have the possibility to issue different texts for each variable that uses the same command group. Thus several variable pairs (each response variable and command variable) use one command group. An additional individualization of the command group is then not necessary.

These texts give the user a better understanding or a better overview of the actions that are available in Runtime (e.g. 'Command: Open disconnecter')

If no limit value has been created for a variable and no reaction matrices are linked, the action uses a standard text:

Switching direction of the action	Standard text
None	@NONE
Off	@OFF
On	@ON
Diff	@INTER
Fault	@FAULT
Direct	@DIR



Information

As the switching direction texts are read out from the limit value settings, they are completely language switchable.

Project overlapping variables



Attention

The variables used in the command groups must be in the same project in order for the command processing to work properly.

If you do use a variable from another project (e.g. subordinate project in multi-project administration), the command processing group, the response variable, the action variable and the action-specific screen ('Command Processing' screen) is expected to also exist in the other project.



Information

You can also use project-overlapping variables for the interlockings by the process. The above limitations apply only to the variables of the command group.

4.2.3 Define Command Processing

Select the **Command Processing** entry in the project tree. Select **New command group** in the context menu.

After creating a new command group, it is added to the detail view of the project manager with standard name "Command group + index". The index is replaced by a consecutive number.

Note: This name serves for the unique identification of the interlocking in the system.



Information

You can assign any name you like to the command groups. However, the names must be unique within the project (applies for standard interlockings and command groups).

The following parameters are available for command groups:

Parameters

Description

Name

Name of the command group. Must be unique among all interlockings in the project. This name is used later with the variable that uses this command group. However the actual assignment is made with a unique, serial numeric ID, so that a command group can be renamed at any time.

Variable name of response

This is the variable name or the mask for the replacement of the response variable.

The placeholder for the replacement text is the character sequence `*` within a name.

Example:

- ▶ `*_RV`
- ▶ `*/stVal[ST]`

Only one placeholder, `*` can be used in the name. When entering the mask, it is important to take care that this name results in an existing variable name after replacement.

Attention: If the name remains empty or the variable that is used here (replaced or absolute) does not exist at the time of compiling, this command group is not available in RT. A corresponding message in the output window points this error out during compiling.

Set status PROGRESS

If activated, status bit **In progress** (PROGRESS) is written for actions `command` and `Manual correction`. The value that the status bit is set to depends on the switching direction of the action.

The status bit is set to 1 if:

- ▶ the **Return state/switching direction** of the action is `ON` or `OFF`.
- ▶ The response variable does not already have the value of the set switching direction.

The status bit is set when checking the interlockings and remains until the execution of the action has been completed.

This also implies that, in the case of **SBO**, the status PROGRESS is only set after a successful 'Select' (`SE+COT_actcon`) and then remains set during watchdog timer or edge delay.

If the execution of the action is triggered by a context menu or if it is a one-step action, the status bit is also set accordingly.

Parameters

Description

Watchdog timer

There is the following setting for this drop-down list:

- ▶ **none:** The watchdog timer (on page 120) is deactivated. However with **SBO**, there is a wait for confirmation of the 'Select' (SE+COT_actcon) and it is then ensured that 'Select' has been ended - so that the PLC reacts to 'Execute' COT_act in the envisaged time. If not, then the 'Select' is deactivated - a 'Cancel' (SE+COT_deact) is sent for this.
- ▶ **Response variable:** The value of the response variable is used to determine if the command was successful.
- ▶ **via cause of transmission:** The Cause of Transmission (COT) of the command variable is used to check whether the process was successful.

Screen modal

If this is active, the screen is displayed modally, independent of the setting 'Modal dialog' in the screen settings.

Screen title from response variable

The identification of the response variable is shown in the screen title. This only happens when there a title was configured for the screen at the frame.

Text is online language switchable.

Screen

Name of the screen to be loaded if the screen is not opened using the 'Screen : Switch to' function.

Note: Actions called up via the context menu open, for the confirmation of the second stage or interlocking text, a screen that was defined for the action. Only if no screen has been linked for the action is a defined screen also used here.

Breaker tripping detection

Only available if property **Set status PROGRESS** is activated.

Active: The response variable is monitored for a change from <> 0 to 0. The identification only sets the status bit CD_TRIP (50) to 1 if:

- ▶ status bit CB_TR_I (51) is not 1, otherwise the identification is suppressed.
- ▶ status bit PROGRESS(10) is not 1, otherwise the value change of the response variable is considered a result of its own command.

Attention: Value changes that are a delayed consequence of its own command can be recognized as breaker tripping. This happens if the PROGRESS bit has already been deleted or if the action does not support runtime monitoring.

Parameters

Suppress detection

Description

Entering the formula with which the detection of a breaker tripping can be suppressed. A click on button ... opens the formula editor.

All variables of the interlocking can be used for the formula. Name replacements with '*' - as with the definition of the interlocking conditions of an action - are possible.

The suppression sets the status bit `CB_TR_I` (51) to 1.

With active recognition, all variables whose status or value are used in the formula for breaker tripping detection are activated for reading when the program is started after loading all projects, and remain this way as long as Runtime is running. Variables from projects loaded in Runtime can be used.

Note: Variables that are used in the formula cannot be deleted from the list of the variables linked to the command group.



Information

The response and action variables do not need to be in the list of the variables linked to the command group. Their names need only be configured for the command group and in the action.

Command Processing in Distributed Engineering



Information

Because the command conditions and the general interlockings (standard functionality / without energy edition) are deposited in the same structure in the editor, the checkout symbol is set to identical for both nodes in the project tree (enable changes). All actions on the command conditions also apply to the general interlockings and vice versa.

Variables marked for deletion are considered as not existent for the compilation of the command conditions. During compiling, the respective error messages are displayed in the output window.

Create action

Actions define the switching commands that are possible for command groups. By selecting the element 'Action' in the detail view of the command group, you can define a new action with a right mouse click. Details of the defined actions are also shown in the detail view after creation (e.g. "switching command: *_BE [ON,1]").

All further settings for the actions are made in the properties window. Some of the properties are inactive, depending on the action type.

Available properties:

Parameters	Description
Action settings	
Action variable	<p>Variables on which is written. For some actions, this is the response variable. In this case, the field is locked.</p> <p>The placeholder for the replacement text is the character sequence <code>,*</code> within a name. Only one placeholder can be used in a name.</p> <p>If the variable that is used here (replaced or absolute) does not exist during compiling, the action is not available in the RT. An according message announces this error during compiling.</p> <p>Click on the <code>...</code> button to open the dialog for selecting a variable.</p> <p>Default: <code>No Allocation</code></p>
Action type	<p>Shows the type of command. For editing, only approved for command action type, where possible settings are <code>switching command</code> or <code>pulse command</code>.</p> <p>Default: <code>Switching command</code></p>
Return state/switching direction	<p>Defines the expected value and the status of the response variable after action execution.</p> <p>Locked for the actions <code>block</code>, <code>lock</code> and <code>release</code>.</p> <p>Default: <code>Off</code></p>
Command	<p>Defines the value that is written to the command variable with the Command action.</p> <p>Note: only available for the command actions <code>switching command</code> and <code>pulse command</code>, <code>auto/remote command</code> and <code>forced command</code>.</p> <p>Default: <code>0</code></p>
Edge delay	<p>Time in milliseconds by which the resetting of the value is delayed for a <code>pulse command</code>.</p> <p>Note: Only available for the <code>pulse command</code> action. There is no wait until until runtime monitoring has ended.</p> <p>Default: <code>1000 ms</code></p>
Set value	<p>Defines the value that is written to the controller.</p> <p>Note: Only available if Return state/switching direction has been set to <code>DIR</code>.</p>
Modifiable states	<p>List of the states which can be modified with the <code>Set status</code> action.</p> <p>Note: Only available for the action <code>'Status input'</code>.</p> <p>Default: <code>None modifiable</code></p>
Command Processing screen	

Screen	<p>Command Processing screen that is used when the action has been carried out using the context menu of the element. If no screen is entered, the screen that is entered in the Screen property for the command is used. An engineered screen which is not available, creates an error message when creating the Runtime files. In this case the action is not taken over.</p> <p>Default: none.</p> <p>Note: If the command processing is called up by a dynamic screen element, this property is ignored and the screen that is entered in the Screen property in the command processing group is always used.</p> <p>Not available for the <code>auto/remote</code> command action type.</p>
Action button	<p>Allocation of the action to an action button in the screen, defined at the command group. If the command group is used for another screen (e.g. via function), the allocation to the action button remains nevertheless. In other words, the action is always placed on the button with the allocated action ID. If such a button is missing, the action is not available in the screen. Only the action buttons that were not allocated yet are provided in the selection list.</p> <p>This setting is locked if no screen was allocated to the command group and for the <code>Lock</code> action type.</p> <p>Default: No Allocation</p>
Nominal/actual value comparison	<p>If this is active, there will be a check whether the value of the response variable already matches the switching direction. If this is true, an unlockable interlocking variable is shown.</p> <p>Note: Only active for <code>command</code> action type</p> <p>Default: Inactive</p>
Close automatically	<p>If this is active then the screen is closed automatically after action execution.</p> <p>Note: Not available for the <code>auto/remote</code> command action type.</p> <p>Default: Inactive</p>
two-stage	<p>If this is active, only after operating the Execute 2 control. Step that executes action. If not active, the action is executed after releasing the last interlocking or, if there is no upcoming interlocking, immediately.</p> <p>Note: Locked for the <code>lock</code> and <code>auto/remote</code> command actions.</p> <p>Default: Active</p>
Two-hand operation	<p>► Active: The Execute 2 control element. Step is only unlocked if the <code>Ctrl</code> key is held down.</p> <p>In Multi-Touch applications, both pressure points must each be on their own screen with their own frame.</p>

	<p>Not available if no two-step execution has been configured.</p> <p>Note: For the Execute 2nd step control element, the selectable with lasso property must not be active with two-hand operation.</p> <p>Default: <i>Inactive</i></p>
Menu ID	<p>The menu ID is used for the creation of Context menus in the Runtime.</p> <p>Note: If two actions are fitted with the same ID, they are tagged with a special symbol in the action tree. They can then not be called up by the context menu.</p>
Options	
Suppress CEL entry	<p>If this is active, no entry in the CEL will be made when executing an action.</p> <p>Default: <i>Inactive</i></p>
Timeout	<p>Timeout for the runtime monitoring in seconds for <code>switching command</code> and <code>pulse command</code> actions.</p> <p>This setting is also applicable as a timeout for <code>Select</code>.</p> <p>Unit is seconds</p> <p>Only available for the actions <code>Command</code>, <code>Auto/Remote command</code>, <code>Check response</code>, <code>Setpoint input</code> and <code>Forced command</code>.</p> <p>Default: 30</p>
Timeout can be canceled	<p>Allows the cancellation of the timeout in runtime monitoring.</p> <p>Only available for the <code>command</code> and <code>Setpoint input</code> actions.</p> <p>If the command has already been executed - after COT_actcon has been received - the Cancel button cancels runtime monitoring.</p> <p>Buttons are therefore active and operable again.</p> <p>Note: Not all drivers support deactivation during execution. If not, no Cancel is sent to the controller; the action is canceled only.</p>
Use Qualifier of Command	<p>Allows commands to provide additional information (Qualifier of Command). The requirement for this is that the driver also supports this option. Possible drivers are, for example, IEC850, IEC870 and DNP3.</p> <p>Is only available for the actions <code>command</code>, <code>auto/remote command</code> and <code>forced command</code>.</p> <p>Default: <i>inactive</i></p>

Qualifier of command	<p>Entry of a numerical value that is sent to the driver as a command parameter. This input possibility is only available if Use qualifier of command has been activated.</p> <ul style="list-style-type: none"> ▶ Entry range: 0 – 127 ▶ Default: 0
-----------------------------	---



Attention

*The identification of the action types in the **menu ID** must be clear, so that they are clearly identifiable in the context menu (on page 90). If two actions have the same ID, they are tagged with the special symbol **M** in the action tree.*



Hint

Note:

- ▶ When selecting individual properties, you receive additional information about functionality in the embedded help.
- ▶ Defined actions and commands can be exported into XML and imported from XML. They can thus be easily archived or reused in other applications.
- ▶ The status can be set using the command **status input**.

Action types

The action types are the available command procedures. According to the command, different activities are performed.

The system provides a variety of actions. The following action types can be defined for the command groups:

Action type	Remark
▶ Command new (on page 78)	Switching command or pulse command. Uses the value of the command variable to write the configured command processing status to the controller. Note: the switching command is suitable for individual and dual commands with the Energy driver (IEC60870, IEC61850, DNP3).
▶ New auto/remote command (on page 79)	The remote command is forwarded from the Process Gateway or the zenon API to the command processing and processed as a switching command. The action is not available in a command processing screen nor via the context menu.
▶ New forced command (on page 80)	The forced command action type allows the setting of a command, even if the response variable is empty, OFF, NT or INVALID . Note: the action is intended for emergency shutdowns and should only be used with caution.
▶ New set value input (on page 80)	Writes a desired numerical value to the command variable.
▶ New status input (on page 81)	Changes the status bits of the response variable. Only applicable for status bits in the modifiable status list.
▶ New replace (on page 82)	Changes the status of the response variable to substitute value (ALT_VAL) and writes value to the response variable.
▶ New revision (on page 83)	Sets the REVISION status bit of the response variable. Note: Alarm handling is suppressed in the revision.
▶ New manual correction (on page 83)	Sets the value of the selected response variable according to the switching direction. Note: the communication protocols in Energy (IEC60870, IEC61850, DNP3) preclude direct writing to the response variable.
▶ New block (on page 84)	Switches off the response variable (OFF status bit). Note: the switched-off variables are no longer read by the connected hardware.
▶ New release (on page 84)	Sets substitute value replacement value (ALT_VAL) to 0. Note: as a consequence, the response variable has received the value from the controller again.
▶ Check response value (on page 85)	Checks the status of the response variable without executing an activity. Note: the action is intended for use in the command sequences module .
▶ New lock (on page 85)	The response variable locks for further action upon entry of a valid locking code .

Note: The action types are listed in the above breakdown in the sequence in which the action types are offered in the zenon context menu. However the sequence in the main window is alphabetical.

In the detail view of command processing, the actions in the tree are shown with the respective selected switching direction and configured action value.



Attention

*The identification of the action types in the **Menu ID** must be clear, so that they are clearly identifiable in the context menu (on page 90). If two actions have the same ID, they are tagged with the special symbol **M** in the action tree.*

Action type command

Depending to the command type, this action type is used in the system as a **switching command** or **Pulse command**.

When the command is executed, a value (0 or 1) is written to the command variable. The value of the command determines the **Command** property of the action.

This action type supports **Select Before Operate** and **Runtime monitoring**. The value which that is then expected for the response variable as a result of the command is to be defined under the **response message status/switching direction** (on/off/none) property.

Switching direction	Value of the response variable after a command has been executed
None	Will not change and must be specifically monitored.
Off	Value will be 0.
On	Value will be 1.

With **pulse commands**, a value is written to the PLC twice. The second time, there is, after the configured **edge delay**, an automatic reset to 0 or 1 (depending on the **switching direction**). However this does not happen if the **Select Before Operate (SBO)** command has been activated for the command variable. If the **SBO** property has been activated for the action variable, a **pulse command** acts in the same way as a **switching command**.

Please keep in mind: The pulse command is not recommended for Energy drivers. The pulse command should only be used with a PLC that expects a pulse instead of an edge.



Note

*If, during the execution of the action, the current value of the response variable is different to the one defined in the switching direction and the switching direction was defined to be `on` or `off`, the **in progress** (**PROGRESS**) status bit is set. To do this, activate the **Set PROGRESS status** property of the command group.*

Auto/remote command action type

The **Remote command** (via Process Gateway, VBA, etc.) is forwarded to the zenon command processing, which processes the sequence (checking of interlocking, forwarding to driver, response, etc.) like a **Switching command** (on page 78) .

The command processing is not accessible via the command screen or the context menu.

The command is only supported by a previous **Select** . The action variable must have the **Select Before Operate** property activated.

When Runtime is ended, or reloaded, any **Select** that has been set is discarded. This means: The master connected to the Process Gateway is not informed and must get this itself using the interruption to the connection.

The VBA interface can use the `IVariable::SetValueWithStatusEx` method and the status bits to be transferred decide whether writing should be either direct or via the command processing. If the status bit `NET_SEL` (bit 8) has already been set (the command processing screen is open for example), the command is not executed. If the status bit is not set, it is set and writing is executed by Command Processing or commands are forwarded to the Command Processing. The response value of the method provides information on whether command processing has been activated or whether the command has been executed.

Transfer of the status bits of the action variable to the method:

- ▶ `SE_870 + COT_act (6)` - Select activation
Determines the command action to be executed and activates the command processing. The response variable of the method provides information on whether this is possible.
- ▶ `SE_870 + COT_deact (8)` - Deactivation (Cancel)
Ongoing command processing is canceled.
- ▶ `COT_act (6)` - Activation (Operate/Execute)
Execute for command Command Processing is executed.

In order for this method to be able to execute command processing, a **remote command** action must exist whose switching direction corresponds to the transferred set value. The actual value written to the driver, **Select** etc, results form the properties of the action.

Note: When an interlocking takes effect, a (language-switchable) CEL entry with the configured text is created.



Information

You can find further information in the Select before Operate chapter in the Process Gateway manual, chapter IEC870 Slave.

Mandatory command action type

The **forced command** action type allows the setting of a command, even if the response variable is empty, OFF, Not topical or invalid (INVALID). It is not intended for emergency shutdowns.

Interlocking conditions cannot be created for the forced command, because it cannot be guaranteed that the condition variables have a valid value in Runtime.

Note: The forced command corresponds to a **switching command** without conditions.



Attention

The early or erroneous configuration of a forced command in Runtime can have dramatic consequences for the equipment. Always set this command with care and protect it with user authorizations.

Action type set point input

The **setpoint input** action type offers the possibility to set any desired numerical value to the command variable. The command processing screen offers special control elements for this, which allow manual definition of the set value. With the help of property **Return state/switching direction** you can define how the set value should be written:

Switching direction	Value of the response variable
DIR	<p>Set value is written directly. You define the value which should be written with the help of function Set value.</p> <p>The text which should be displayed can be engineered using a limit value/rema for the state/value 5. If this is not the case, a standard text (on page 66) is used.</p> <p>Nominal/actual value comparison is not supported.</p> <p>The action can be carried out several times in a row.</p>
Set value	<p>Value of the command processing screen of the Set value element is written to the action variable.</p> <p>In one-step execution, the value is written when pressing the Execute standard key or when pressing the action button (if configured).</p> <p>In two-step execution, the value is written when pressing the Execute 2 step standard key.</p>



Information

If a variable is linked to a command group, it is not possible to describe the variable with the zenon **Write set value** function.

*Exception: If a write set value (on page 80) command with switching direction set value has been created, the zenon function calls up this action in the background without the command input screen being called up. This means that the **command conditions** (on page 99) are checked. An active interlocking condition prevents the writing of a set value. During the execution of an action, the `NET_SEL` status bit is not set and the **Select Before Operate** variable property is ignored.*

*This is also applicable for the value entry of a variable that is linked to a dynamic element if `Element` was selected for the **Write set value via** property.*

For further information, read the information in the Apply actions (on page 87) chapter.



Attention

When writing the set value with the switching direction `DIR`, neither the limits of the linked variable are checked, nor is a check carried out to see whether write set value is permitted for this variable.

Action type status input

Changes the status bits of the response variable. The following is executed, depending on the definition of the **switching direction**:

Switching direction	Action
Off	The states configured in the Modifiable states list are all reset to 0.
On	The states configured in the list Modifiable states are all set to 1 (active).
None	The states configured in the Modifiable states list must be defined in Runtime in the Command Processing screen with the help of the <code>Set status</code> control element. Each status bit is defined individually using a checkbox in the control element.

If you change a status bit in Runtime, the change is logged in the Chronological Event List (status including value). These language of these messages can be switched in Runtime.



Information

For all status defaults, there is always a write to the response variable.



Info

If a switch is locked using the `Lock` action, the status bit `M1` of the response variable is set.

The status bits `OFF` and `REVISION` are also handled in other actions.

*In addition, the status bits `CB_TRIP` and `CB_TR_I` reflect the results from the property of the **breaker trip detection command group**.*

Action type replace

The process value of a remote-controlled switch is temporarily replaced with a replacement value (due to revision, maintenance work, or an ongoing connection outage, for example).

The response variable is set to the status alternative value **Alternate value** (ALT_VAL). In addition, the value defined by the **switching direction** is placed on the response variable.

Switching direction	Alternate value
Off	0
On	1
Diff	2
Fault	3
None	4

Action type revision

Set or reset the status bit Revision of the response variable as engineered in the property **Switching direction**

Switching direction	Status
Off	Set to 0
On	Set to 1

Action type manual correction

The **correct** action sets the value of the response variable according to the setting of the **switching direction**:

Note: the communication protocols in Energy (IEC60870, IEC61850, DNP3) preclude direct writing to the response variable. The action will be unsuccessful in these drivers! To execute a command, the setting of the value to a command variable is expected.

Switching direction	Action
Off	0
On	1
Diff	2
Fault	3
DIR	<p>The set value is written directly. You define the value which should be written with the help of function Set value.</p> <p>The text to be displayed can be configured using a limit value or a reaction matrix for the state/value 5. If this is not the case, a standard text (on page 66) is used.</p> <p>Nominal/actual value comparison is not supported. The action can be carried out several times in a row.</p>
Set value	Value of the Set value control element is written to the response variable in the Command Processing screen.



Attention

When writing the set value directly neither the limits of the linked variable are checked nor is it checked if the write set value is allowed for this variable.



Information

*The **In progress (PROGRESS)** status bit is set if:*

- ▶ When the action is carried out, the current value of the response variable is different to the value set for the **switching direction** and
- ▶ the **switching direction** was defined as `on` or `off`.

MANUAL CORRECTION

Manual correction is the manual correction of a non-remote switch in zenon. A variable is usually corrected without a connection to the process. There should never really be an invalid i-bit pending for such variables. It is indeed possible, but it makes no sense to correct a variable with a reference to the process! The PLC will overwrite this value again.

Behavior:

Correction is completely normal value setting from the perspective of the driver.

Opposite of this - Action: Replace (on page 82)

The process value of a remote-controlled switch is primarily replaced with a replacement value (due to maintenance work, for example).

Action type block

the response variable is switched off as a result of executing this action.

The status bit of the response variable is set to OFF. The switched-off variables are no longer read by the connected hardware.

Note: Can only be configured once per command group.

Action type release

The **Release** actions resets the **replacement value (ALT_VAL)** status bit to 0 (inactive). If the **Switched off (OFF)** status bit is also active, it is also set to 0 (inactive). runtime receives the current value from the driver for the response variable once the **Release** action has been carried out.

The action can only be executed in Runtime, if the **replacement value (ALT_VAL)** (value: 1). is active for the selected response variable. 1).

Note: Can only be configured once per command group.

Check response value action type

The **Check response value** action type is to check variables for the status **ON** or **OFF**.

Whilst the **Check response value** action is executed, the standard key **Cancel** is unlocked in the Command Processing screen.

In doing so - depending on the setting of the **runtime monitoring** (on page 124) - there is a wait until the value of the response variable corresponds to the value of the checking direction - **switching direction** action property. If the checking value is **Ein**, this is the value 1; it is the value 0 for **OFF**.

If no runtime monitoring has been configured (**runtime monitoring**= "none"), the set waiting time (~24 hours) is the maximum time that is waited. Otherwise the action is ended and the **TIMEOUT** status bit is set for the response variable.

If, after execution of the action in the command processing screen, the other actions are not available, this is for the following reasons:

- ▶ The **timeout** for **runtime monitoring** has not yet expired.
- ▶ The response variable does not yet have the expected value (the value change has not yet been received).
- ▶ The action has not yet been canceled with the **Cancel** button.



Information

*The **Check response value** action only serves to read the value of the response variable without executing an activity.*

*The action is intended for use in the **Command Sequencer module**.*

If the response variable already has the value of the **switching direction**, the execution of the action is recognized as completed. The other buttons in the command processing screen are thus immediately available.

Note: If the response variable is set to **OFF** or **Revision**, the response value can nevertheless be checked.

Action type lock

Enables the lock of a response variable for the actions of the command processing.

Note:

- ▶ Can only be configured once per command group.

- **Interlocking conditions** are not supported for the action. Locks can always be executed.



Information

If a switch is locked using the `Lock` action, status bit M1 is set.

A prerequisite for this is that users have a **lock code** configured in the **user administration module**. Locking or unlocking a response variable can only be done with the correct input of a **lock code**.

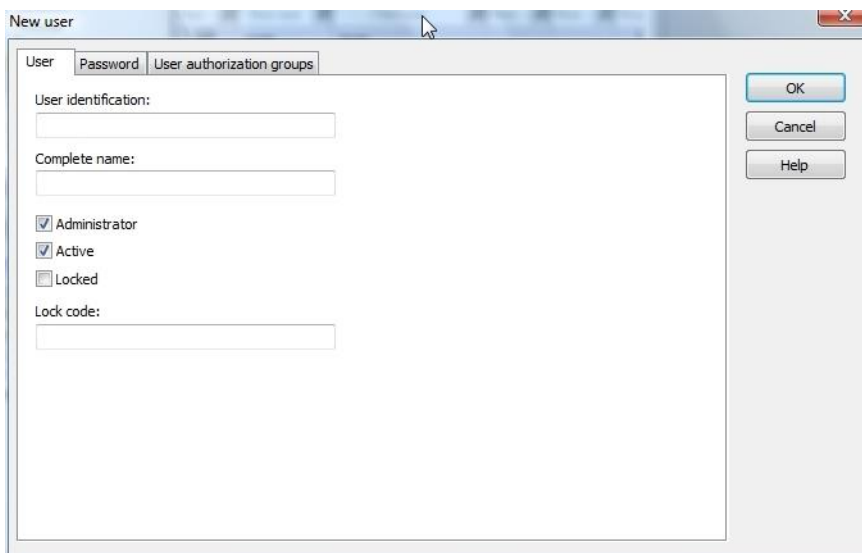
The same variable can be locked by multiple users in parallel. Actions for the response variables are possible only after all locks have been unlocked by entering the **lock code**.

There can be no actions executed if

- Actions of the command variables use the locked variable as response variable (e.g. **switching command**)
- Actions of the command variables use the locked variable as an action variable (e.g. **replace**)

A list of the currently-active locks can be shown in the command processing screen using a special **lock list** control element.

The **lock code** can be defined individually for every user. These parameters are set directly for a pre-existing user with the **lock code:** property.



You can also set the **lock code** for an existing user in Runtime.

In the Runtime you cannot delete users who still have an active command lock.



Attention

Users can be deleted in the development environment. This causes the loss of the defined locks after restarting or reloading.

Users locked (activated) in the **user administration** cannot activate or deactivate command locks.



Information

Information about active locks is also synchronized in the redundant network and is therefore available after redundancy switching.

Apply actions

Command Processing in the Energy Edition can be used in different situations. The user can choose the variant they prefer. A simultaneous use (related to an element) of the different types of use is possible at any time:

- ▶ Calling up a **screen switching** function on a **Command Processing** screen (on page 88).
- ▶ Calling up a **numeric value, combined element, dynamic text, bar graph, clock, universal slider, pointer instrument** or **status element** screen.
For activation, the **Write set value** property must be configured to `Command Processing` using the element.
- ▶ Call via a context menu if `Command Processing` was set for the **action type** property. The command processing screen is opened for any possible interaction with the user (e.g. pending interlocking).
- ▶ It is called up using the **Command Sequencer module**.

As soon as the variable is linked to a command group, direct input of set values is only possible using a command processing screen or a `command` context menu. Exception: If the command group contains a **setpoint input** action with **switching direction** 'set value', this action is used for the command variables (not response variables).

This happens:

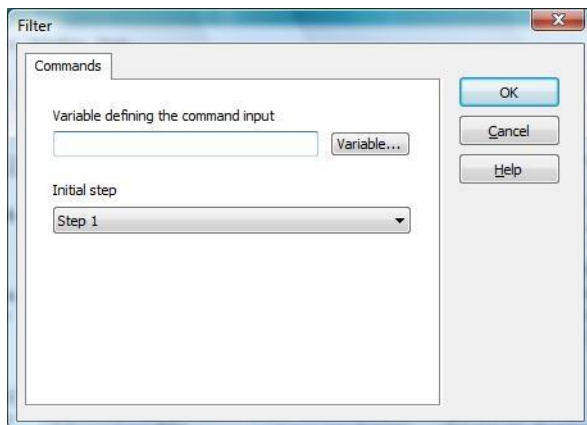
- ▶ When calling up the **Write set value** function.
- ▶ When calling up the screen element; also if the **Write set value via** of the element has the value `dialog box` or `element`.
- ▶ When calling up a `set value` context menu.

However then neither the status bit `NET_SEL` of the response variable is taken into account nor is a `Select` executed.

Attention: with this type of execution, a pending interlocking condition in the **write set value** action prevents writing of a set value. In doing so, no interaction with the user is undertaken.

Screen switch to Command Processing screen

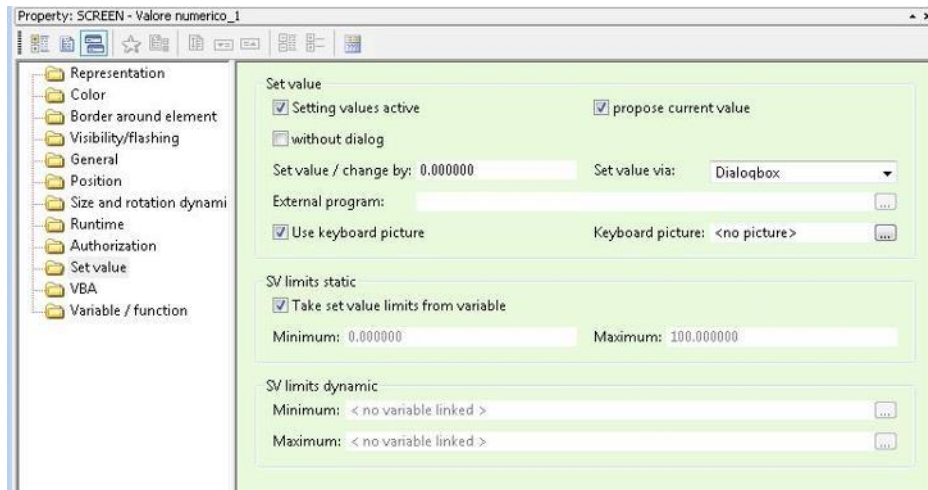
If a `Command Processing` screen is selected with the **screen switch** function, the configuration dialog for the screen switching function has the following parameters:



Parameters	Description
Variable defining the Command Processing	The variable configured here defines the command group to be used. The screen determines the appropriate response variable and the associated action variable via the name of the variable.
Initial step	<p>Defines the step (status) in which the command processing screen is loaded.</p> <ul style="list-style-type: none"> ▶ Step 1 The screen is loaded and waits for action definition and action execution. Action executions must be performed manually by the user. ▶ Block The screen is opened in the command step for the action block. Note: Note that not all configured control elements are visible with this initial step. You can find an overview of all visible control elements in the blocked or locked elements (on page 159) chapter.

Command processing via dynamic screen elements

If, for a dynamic screen element, the `Command Processing` option is configured for the **Set value via** setting, the command window can be opened in Runtime by left-clicking on the element. The screen to be opened is defined at the command group of the variable linked to the element. The corresponding action or response variable is automatically determined from the response variable.



Information

With the 'Command Processing' option selected, the command processing screen is called up instead of the standard 'Write set value' standard dialog.

The following screen types support the `Command Processing` option:

- ▶ Numerical value
- ▶ Combined element
- ▶ Text element
- ▶ Bar display
- ▶ Pointer instrument
- ▶ Status element
- ▶ Clock
- ▶ Universal slider

If no command group was defined for the variable assigned to the element, or if the response variable of the command group does not exist, an error entry for the **diagnosis viewer** is generated. The screen is not opened then.



Information

If a variable is linked to a command group, it is not possible to describe the variable with the zenon **Write set value** function.

*Exception: If a write set value (on page 80) command with **switching direction set value** has been created, the zenon function calls up this action in the background without the **command input** screen being called up. This means that the **command conditions** (on page 99) are checked. An active interlocking condition prevents the writing of a set value. During the execution of an action, the **NET_SEL** status bit is not set and the **Select Before Operate** variable property is ignored.*

*This is also applicable for the value entry of a variable that is linked to a dynamic element if **Element** was selected for the **Write set value via** property.*

Command processing via context menu

The command processing can also be instigated at the element directly via the context menu - property **Runtime - context menu**. This is the most frequently used method. In this regard, the context menu is already the first step of the two-step action.

The menu must have an entry of the **command processing** action type. The display of the single action is defined automatically by the menu. The display of the actions can be influenced selectively, depending on the '**names**' of the menu entry.

When creating a new action in the Command Processing (on page 57), a menu ID corresponding to the action type and the switching direction for the **Action type** property is created and offered in the drop-down list. If the content corresponds to an ID defined as standard text for the action type and switching direction, the content is adapted if the action type or switching direction change.

To create a context menu for the Command Processing:

1. Create the desired actions in the command processing (on page 57)
2. In the properties of the context menu item select the **Action type Command Processing**
3. Select the desired action and switching direction via the drop-down menu with the **Menu ID** property
4. Give it a clear label in the **Text** property
 Note: If no entry is defined for **Text**, the field is automatically filled with the "**command processing**" label.



Attention

The name of the *Text* property must be unique. If two names the same are issued, further menu items with the same name are not displayed.

Because automatically created menu items with the same action result in the same text, there are macros (on page 94) available for these.

The character sequence `ID_CMD_AUTO` is reserved for automatically created menu items. These must always be used with macros, because otherwise only the menu item is inserted.

When checking for duplicate entries the following rules apply:

- ▶ Manual menu points have priority over automatic ones.
- ▶ If it is the same type then the last entry has twice the priority.
- ▶ If a duplicate entry is found, a warning is set off in the log. This includes the menu ID and description. Automatically expandable entries have **<auto>** added to the ID.

ACTIONS FOR ACTION TYPE COMMAND PROCESSING

Action	Switching direction	Menu ID
ID_CMD_AUTO		This menu entry automatically shows all possible actions for an element, if no direct menu entry from the list is used already.
Pulse command	On (1)	ID_CMD_EBEF_ON
Pulse command	OFF (1)	ID_CMD_EBEF_OFF
Pulse command	NONE	ID_CMD_EBEF_NONE
Switching command	On (1)	ID_CMD_DBEF_ON
Switching command	OFF (2)	ID_CMD_DBEF_OFF
Switching command	NONE	ID_CMD_DBEF_NONE
Set value	NONE	ID_CMD_SVALUE
Set value	DIRECT	ID_CMD_SVALUE_DIR
Status input	NONE	ID_CMD_STATE
Status input	On (1)	ID_CMD_STATE_ON
Status input	OFF (0)	ID_CMD_STATE_OFF
Replace	NONE	ID_CMD_REPL_NONE
Replace	On (1)	ID_CMD_REPL_ON
Replace	OFF (0)	ID_CMD_REPL_OFF
Replace	DIST	ID_CMD_REPL_DEF

Replace	DIFF	ID_CMD_REPL_DIFF
Manual correction	NONE	ID_CMD_UPD_NONE
Manual correction	On (1)	ID_CMD_UPD_ON
Manual correction	OFF (0)	ID_CMD_UPD_OFF
Manual correction	DIFF	ID_CMD_UPD_DIFF
Manual correction	DIST	ID_CMD_UPD_DEF
Manual correction	DIRECT	ID_CMD_UPD_DIR
Block	NONE	ID_CMD_BLOCK
Release	NONE	ID_CMD_UNLOCK
Lock	NONE	ID_CMD_LOCK
Revision	OFF (0)	ID_CMD_REV_OFF
Revision	On (1)	ID_CMD_REV_ON
Mandatory command	On (1)	ID_CMD_FORCE_ON
Mandatory command	Off (0)	ID_CMD_FORCE_OFF
Mandatory command	NONE	ID_CMD_FORCE_NONE

NAME OF THE MENU ITEMS OF THE CONTEXT MENU

1. AUTOMATIC CREATION

Entries that were created using ID_CMD_AUTO automatically get a name according to the following pattern: 'Action name' plus 'Limit value text of the switching direction'.

2. MANUAL CREATION FROM TABLE

If the menu entries are created from the table, for every action under 'Display - Text' a text must be defined for the entry in the context menu.

Names for the menu entries:

Command, Set value, Status, Replace, Release, Manual correction, Block, Lock, Revision

ACTION TEXTS

Action	Text
Pulse command Switching command	Text from the limit value text, according to the switching direction.
Manual correction Replace	If a switching direction (other than 'None') is defined, the text from the limit value text according to the switching direction is displayed.
Status	'OFF' or 'ON', depending on the set switching direction
Revision	Text from the limit value text, according to the switching direction.
Others	No special action text is displayed.



Example

Displayed text for a switching command with defined limit value:

'Command: switching direction ON'



Information

- ▶ All displayed texts are language switchable with the standard mechanisms.
See also: Which texts are language switchable?
- ▶ All displayed menu entries are automatically sorted alphabetically.

The currently used command group is determined via the variable which is linked with the screen element. If no command group is assigned to the variable or if there is no response variable, the context menu is not displayed in the Runtime (an according error message is transferred to the diagnosis server).



Information

*The menu entries of the command processing are displayed depending on the command group. The menu entry is showed only when the connected action exists. Consequently, if the variable of the element is the command variable, only **the** actions for the command variable plus the action 'Lock' can be displayed. Actions for the response variable are hidden automatically.*

AVAILABILITY CONDITIONS

The menu entries are only released when the corresponding actions are executable. The following conditions are requirements:

- ▶ All menu entries are locked if the `NET_SEL` status bit of the response variable is active.
- ▶ All menu entries are locked, when the response variable could not be determined.
- ▶ All menu entries are locked, when the response variable has no value and could not get a value within 30 seconds.
- ▶ All menu entries are locked on an Internet Client without write access.
- ▶ Menu entries are locked when there is no connection to the server.
- ▶ The menu entry connected to the 'Release' action is locked when the `ALT_VAL` status bit of the action variable is not active.
- ▶ The menu entry connected to the 'Replace' or 'Revision' action, whose switching direction matches the value of the action variable, is locked.
- ▶ All menu entries, except the one which is connected with the action 'Lock', are locked, when a change lock is active for the response variable.
- ▶ When the `REVISION` status bit of the response variable is active, the actions 'Set value', 'Replace', 'Correct', and 'Command' are locked.
- ▶ As long as a watchdog timer, an edge generation or an SBO is active for the command group, all menu entries are locked. This results from the fact that the `NET_SEL` status bit also stays active.

Macros for the context menu

A macro is a defined character sequence that is replaced by another text when menu items are created in Runtime. Virtually all macros can occur more than once per menu item. They can also contain further macros as a result. In doing so, the expansion sequence must be considered. Macros are not case sensitive when configuring menus. If macros contain a macro as a result, the macro must be contained in capitals in the result. The entry is made with `$` as a prefix and suffix.

The sequence of the expansion is from left to right in the following priority.

1. **\$ALL\$**
2. **\$NOTE\$**
3. **\$TAG\$**
4. **\$REMA<Status>\$**
5. **\$RDIR\$**
6. **\$DIR\$**
7. **\$ACT\$**
8. **\$NAME\$**

Macro	Description
\$ALL\$	<p>Results in Action naming: Switching direction.</p> <p>Corresponds to the combination of the \$ACT\$ macro \$DIR\$</p> <p>Note: If a context menu is created for the command processing, the default text is \$ALL\$, even if the menu already has text configured for it but the action type changes to command processing.</p>
\$NOTES\$	<p>The whole text including the macro is interpreted as a note. If the resulting text is empty, the \$ALL\$ macro is used.</p> <p>For the last macro, the note macro is again checked and the text to the right of this including the macro is deleted.</p> <p>If the resulting text is empty or only consists of spaces, the menu item is not inserted.</p>
\$TAG\$	<p>Is replaced by the identification of the action variable.</p> <p>The identification can be translated by the online language translation function. If no translation character (@) is contained, the whole identification is highlighted for translation.</p>
\$REMA<Status>\$	<p><Status> is a Rema or limit value state, the text of which is used as a replacement.</p> <p>If the status is not present, the menu item is not displayed.</p> <p>The limit value text is translated linguistically according to the placement of @ .</p> <p>The status can be a number between -2^{31} and 2^{31-1}. Leading characters and a prefix are permitted. If characters are contained that cannot be converted to a number, or the number is outside the given area, the menu item is not displayed.</p>
\$RDIR\$	<p>Text for the switching direction from reaction matrix/limit value as in \$DIR\$ macro, with the exception of:</p> <ul style="list-style-type: none"> ▶ <code>Action Write set value direct</code> The text is taken from the rema/limit value of the status, which corresponds to the value of the set point to be set. ▶ <code>Action Status on and Status off</code> Text is taken from the rema/limit value for the <code>on</code> or <code>off</code> statuses. ▶ <code>Action Correct direct</code> The text is taken from the rema/limit value of the status, which corresponds to the value of the set point to be set.
\$DIR\$	Switching direction of the action.
\$ACT\$	Action naming of the action.
\$NAME\$	The \$NAME\$ macro can be used to create menus and provides the configured content of the Action name property, the language of which can also be switched in Runtime with a @ character.

AUTOMATICALLY CREATED MENU ITEMS

Automatically created menu items are created as a menu ID with ID_CMD_AUTO. In this case, macros must always be used, because otherwise only a menu item would be inserted.

COMPATIBILITY

Previous to version 6.51 text at automatic menu items was ignored. When converting projects that were created with versions earlier than 6.51, the macros \$ALL\$NOTE\$ are automatically inserted before the configured text. Therefore the menu items behave as before.

ONLINE LANGUAGE SWITCH

The labeling for the menu item in the **Text** property is translated linguistically before macro expansion from the character @.

Note: If, for the **\$TAGSS** macro, no translation indicator (@) is contained, the complete text is translated.

Error messages when the context menu is called up

When menus are loaded in the Runtime environment, their content is checked for consistency. If an error occurs, corresponding error messages are issued for the **Diagnosis Viewer**. The following messages can appear:

Parameters	Description
Menu entry for command processing suppressed, because name is several times in the menu!	The menu already contains a menu entry with the name used in the command processing. Do not use that name for any other menu entries for the command processing.
Menu entry for command processing suppressed, because description is several times in the menu!	There is already a menu entry with the same description in the menu. Automatically created menu entries are not added, when a menu entry with the same description is already there.
Text for menu entry cannot be detected!	The description of an automatically created menu entry could not be determined. This most probably indicates a missing limit value text.
No command group linked to variable of the screen element!	The variable associated with the screen element has no command group or a no longer valid command group. According error messages are given during compiling.
Response variable does not exist!	The response variable used in the command group does not exist.
Select cannot be activated!	Status bit <code>NET_SEL</code> (8) could not be activated within the timeout.

Execution of actions via the context menu

After activation of a menu item for command processing, the assigned action is carried out. Execution via a menu activates the setting of the `NET_SEL` status bit in the first step. Only if this is successful is the execution of the actual action ('switching command', for example) started.

A command processing screen is then opened if one of the following criteria has been met:

- ▶ If the action to be executed is **Write set value**, **Status input** with input or **Correction**, the screen assigned to the action in the "Stage 1" step is opened. The set value /status to be written can then be defined in the screen.
- ▶ If the action to be carried out is **lock**, the action-specific screen is called up with the "lock" step.
- ▶ If an active locking condition prevents execution, the screen configured in the **Unlocking** step for the action is called up. Execution is also prevented if SBO could not be activated without errors.
- ▶ If two-stage execution is configured for the action, the action-specific screen is called up in the "Stage 2" step.
- ▶ If no specific screen has been configured for the action, the screen that has been configured centrally for the command group is opened.



Information

If none of the above-mentioned conditions are applicable, the action is executed immediately, without further operations.

Set value context menu

If the variable assigned to a screen element is linked to a command group, the writing of a set value is also handled by the command processing. The requirement for this is that a `Write set value` action is present with `set value switching direction` in the Command Processing. If this is missing, the writing of the set value is not carried out.



Information

An active interlocking condition prevents the writing of a set value.

Command conditions

Command groups contain both the definition of the switch actions and the definition of the command interlocking conditions. Command conditions are optimum parameters that can be defined application-specifically.

Each action within a command group can also be supplemented with **command conditions**. These process-controlled interlockings prevent unwanted execution of actions, depending on the current process state.

The following three parts are significant in a command group for the command conditions:

Parameters	Description
Actions	These define which command is executed, on which variables these actions are applied and set the parameters for the internal interlockings.
Condition variables	These define which variables in the command conditions can be used.
Command conditions	The execution of commands in is also made dependent on the current process status by these conditions.

The actions automatically check the internal conditions. These internal conditions are to check whether the basic requirements for the execution of an action have been met (plausibility check). Example: The activation of the `Select` will be rejected by the PLC.

Note: Configured interlocking conditions have no influence on this check.

See also

General interlockings

Create command conditions

Any number of command conditions can be defined for every action. These conditions allow for an additional restriction of the ability to execute an action. These conditions are defined with formulas, in which you can use the variables from the active projects. The formula addresses the linked variables via the index in the condition.



Information

The condition variable is automatically replaced if a '' is used in the definition.*

Define condition variable

First we have to define variables which can be used later for the formulas of the command conditions. If the defined conditions are fulfilled by the linked process variables later, during Runtime, the user has the respective actions available.



Information

If a variable which was used in a command condition is deleted later on, the index within the condition is adjusted. The succeeding variables are put forward and the formulas are adjusted automatically.

The following procedure is recommended for defining a command condition:



Example

1. Select the node Variable in the detail view of the Command Processing and select the option 'New' in the context menu.
2. In the selection dialog, select a process variable, which serves as the base for the formulas of the command conditions. You can also abort the variable selection dialog, which leads to an empty definition. You can define an automatic replacement for this empty link with a '*'.
3. Select an already existing action and the node 'Conditions' in the detail view. With 'New condition', you can define any number of conditions for every action. The definition is performed with formulas; non-fulfilled conditions cause a lock of the associated actions in Runtime.

Interlocking condition

Any number of command conditions can be defined for every action. These conditions are checked before execution of the respective action. If a check fails, the respective action cannot be executed in Runtime.

The conditions are defined as formulas. This syntax is analogous to the definition of the formulas in the Formula Editor.

Additionally, the following interlocking types (in addition to the command groups) are checked before action execution:

INTERNAL INTERLOCKING CONDITIONS

These conditions are checked automatically before every action execution; the engineer cannot influence this. These Internal interlocking conditions (on page 102) are predefined by the system and serve as plausibility checks.



Example

- ▶ The response variable is already selected in the zenon network (has set `NET_SEL` status from other Client).
- ▶ The response variable already has the desired value and the action was configured with **Nominal/actual comparison**.
- ▶ In the SBO, the Select was not accepted by the controller (status bits: `SE_870 + COT_actcon(7) + N_CONF`).

TOPOLOGICAL INTERLOCKING CONDITIONS

These conditions result from the current topological status during Runtime. These conditions are defined in the 'Configuration of the topological interlockings (on page 32)' settings.

Internal interlocking conditions

With the help of the internal interlocking conditions the basic requirements for the action are checked (plausibility check). The results or the addressing of an interlocking are displayed in Runtime in the command processing screen as interlocking text.

Parameters	Description
Status already exists	<p>The state which should be set equals the current value of the response variable. This check is only active if the command group of the 'Nominal/actual value comparison' is active.</p> <p>This interlocking can be unlocked (insofar as the user has authorization for this).</p>
Internal error occurred	<p>Command Processing cannot execute the check.</p> <p>This happens when the data type of the action variable is not allowed for this action.</p> <p>Example: 'Pulse command on' action for string variables.</p> <p>This interlocking cannot be unlocked.</p>
no interlocking object	<p>Command group could not be determined.</p> <p>This interlocking is not unlockable.</p>
Action not defined	<p>Action to be executed could not be determined.</p> <p>This interlocking is not unlockable.</p>
Differences between local and global interlocking	<p>Pulse command parameter not consistent. Error setting parameters</p> <p>This interlocking cannot be unlocked.</p>
One or more values are not available	<p>Value of condition variable not available. Lock code: 14</p> <p>Value of condition variable invalid (INVALID). Lock code: 15</p> <p>This interlocking is not unlockable.</p>
Locking administration not valid	<p>The administration of the lockings could not be loaded or is invalid.</p> <p>This interlocking is not unlockable.</p>
Variable locked for changes	<p>Command Processing locked by response variable (status bit M1).</p> <p>This interlocking is not unlockable.</p>
SBO rejected	<p>The activation of the Select was rejected by the PLC.</p> <p>This interlocking is not unlockable.</p>
Timeout for SBO activation	<p>No confirmation for the activation (positive or negative) was received within the timeout.</p> <p>This interlocking is not unlockable.</p>
Timeout for SBO deactivation	<p>No confirmation for the deactivation (positive or negative) was received within the timeout.</p> <p>This interlocking is not unlockable.</p>
Timeout for execution	<p>There was no notice for finishing the action execution within the timeout.</p> <p>The TIMEOUT status bit is set for the response variable.</p> <p>This interlocking cannot be unlocked.</p>
SBO expired	<p>The PLC has reported the expiration of the SBO activation. The second execution step will attempt to send a Select again.</p> <p>This interlocking cannot be unlocked.</p>

Formula editor

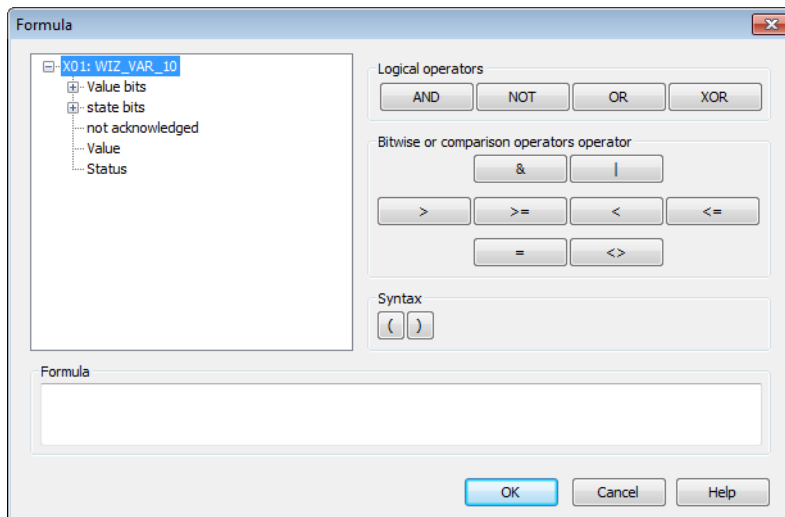
The formula editor provides support when creating formulas with logical or comparative operators with a combined element, for interlockings and command processing. If additional variables are required for a formula, create these in the **formula variables** area of the status window by clicking on the **Add** button. existing formulas are displayed in the status list with the letters **F**.

Note on the input of decimal points:

- ▶ Decimal separator: Comma (,) is automatically converted into a dot (.):
- ▶ Zero as a decimal point is removed automatically; 23 , 000 automatically becomes 23

CREATING A FORMULA

Click on the **Formula** button in the status window The formula editor opens



You select the bits for your formula in the left screen.

On the right, you find the operators for logical and comparative operations.

The formula created is displayed in the **Formula** area.



Information

Up to 99 variables can be linked in one formula. X01 to X99. The length of the formula must not exceed 4096 characters.

THE MEANING OF THE BITS:

Parameters	Description
value bits	32 value bits (from 0 -31) are available. They describe the variable value bit by bit. For binary variables, only bit 0 is of importance, for SINT and USINT only the bits from 0-7, etc. Note: The value refers to the raw value (signal range) of the variables and not to the converted measuring range.
State bits	Here you find the most commonly used status bits. You find the exact definition and use of the status bits in the Status Bits List (on page 107).
unreceipted	Not acknowledged is treated like a usual status bit. But here it is listed separately, because it does not belong to the classical variable statuses.
value and status	In the formulas, all values (value bits and status bits) are treated as binary values and can be logically linked with AND, OR, etc. The total value and overall status are an exception to this. In order to get a Boolean result this total value has to be ORed with a constant bitwise (on page 110). For this, we use the operator &. For the result 0 (false) of this logical ORing we get the binary value 0 (false), otherwise 1 (true). Example: see chapter Example bit by bit ORing (on page 110)



Info

The status bits NORM and N_NORM are only available in the formula editor and cannot be engineered via the status.

If other settings outside the formula are set for the current status, they are combined with the formula with a logical AND.

Refer to the examples (on page 112) section for examples.



Information

Formulas with binary X values and bitwise linking can be used with a maximum of 2 binary values. If more values are required, the linking must be carried out without binary X values.

Example:

X01.Value & X02.Value -> works

X01.Value & X02.Value & X03.Value -> does not work

But:

X01.00 AND X02.00 AND X03.00 AND X04.00 AND X05.00 -> works

List of status bits

Bit number	Short term	Long name	zenon Logic label
0	M1	User status 1; for Command Processing: Action type "Block" (on page 85); Service Tracking (Main.chm::/IEC850.chm::/117281.htm) of the IEC 850 driver	_VSB_ST_M1
1	M2	User status 2	_VSB_ST_M2
2	M3	User status 3	_VSB_ST_M3
3	M4	User status 4	_VSB_ST_M4
4	M5	User status 5	_VSB_ST_M5
5	M6	User status 6	_VSB_ST_M6
6	M7	User status 7	_VSB_ST_M7
7	M8	User status 8	_VSB_ST_M8
8	NET_SEL	Select in the network	_VSB_SELEC
9	REVISION	Revision	_VSB_REV
10	PROGRESS	In operation	_VSB_DIRECT
11	TIMEOUT	Runtime exceedance	_VSB_RTE
12	MAN_VAL	Manual value	_VSB_MVALUE
13	M14	User status 14	_VSB_ST_14
14	M15	User status 15	_VSB_ST_15
15	M16	User status 16	_VSB_ST_16
16	GI	General query	_VSB_GR
17	SPONT	Spontaneous	_VSB_SPONT
18	INVALID	Invalid	_VSB_I_BIT
19	T_CHG_A	Daylight saving time/winter time announcement	_VSB_SUWI
20	OFF	Switched off	_VSB_N_UPD
21	T_EXTERN	Real time external	_VSB_RT_E
22	T_INTERN	Realtime internal	_VSB_RT_I
23	N_SORTAB	Not sortable	_VSB_NSORT
24	FM_TR	Error message transformer value	_VSB_DM_TR
25	RM_TR	Working message transformer value	_VSB_RM_TR
26	INFO	Information for the variable	_VSB_INFO

27	ALT_VAL	Alternate value If no value was transferred, the defined alternate value is used otherwise the last valid value is used.	_VSB_AVALUE
28	RES28	Reserved for internal use (alarm flashing)	_VSB_RES28
29	N_UPDATE	Not updated	_VSB_ACTUAL
30	T_STD	Standard time	_VSB_WINTER
31	RES31	Reserved for internal use (alarm flashing)	_VSB_RES31
32	COT0	Cause of transmission bit 1	_VSB_TCB0
33	COT1	Cause of transmission bit 2	_VSB_TCB1
34	COT2	Cause of transmission bit 3	_VSB_TCB2
35	COT3	Cause of transmission bit 4	_VSB_TCB3
36	COT4	Cause of transmission bit 5	_VSB_TCB4
37	COT5	Cause of transmission bit 6	_VSB_TCB5
38	N_CONF	Negative confirmation of command by device (IEC 60870 [P/N])	_VSB_PN_BIT
39	TEST	Test bit (IEC870 [T])	_VSB_T_BIT
40	WR_ACK	Writing acknowledged	_VSB_WR_ACK
41	WR_SUC	Writing successful	_VSB_WR_SUC
42	NORM	Normal status	_VSB_NORM
43	N_NORM	Deviation normal status	_VSB_ABNORM
44	BL_870	IEC 60870 Status: blocked	_VSB_BL_BIT
45	SB_870	IEC 60870 Status: substituted	_VSB_SP_BIT
46	NT_870	IEC 60870 Status: not topical	_VSB_NT_BIT
47	OV_870	IEC 60870 Status: overflow	_VSB_OV_BIT
48	SE_870	IEC 60870 Status: select	_VSB_SE_BIT
49	T_INVALID	Time invalid	not defined
50	CB_TRIP	Breaker tripping detected	not defined
51	CB_TR_I	Breaker tripping detection inactive	not defined
52	OR_DRV	Value out of Range	not defined
53	RES53	reserved	not defined
54	RES54	reserved	not defined

55	RES55	reserved	not defined
56	RES56	reserved	not defined
57	RES57	reserved	not defined
58	RES58	reserved	not defined
59	RES59	reserved	not defined
60	RES60	reserved	not defined
61	RES61	reserved	not defined
62	RES62	reserved	not defined
63	RES63	reserved	not defined



Information

In formulas all status bits are available. For other use the availability can be reduced.

You can read details on status processing in the Status processing chapter.

Logical operators

Logical links: Variables will only be checked for the logical value '0'; if the value does not equal '0', it will be considered as '1'.

In contrast to bit formulas, the technical range can be modified by a stretch factor -> (not equal '0' or '1').

Operator	Meaning
AND	logical 'AND'
NOT	Negation
OR	logical 'OR'
XOR	logical 'EXCLUSIVE OR'

The operators have the following priority in the formula calculation:

Priority	Operator
1	& (operator for bit formulas (on page 110))
2	NOT
3	AND
4	XOR/OR

**Info**

Up to 99 variables can be linked in one formula. X01 to X99.

**Info**

The status bits NORM and N_NORM are only available in the formula editor and cannot be engineered via the status.

Bit formulas

Bit formulas only have a logical high or low state. In contrast to logical formulas, the raw value is already predefined (0,1).

Operator	Description
&	AND
	OR

Example: ORing bitwise

You want to find out if one of the user status bits 1-8 (M1 ... M8) of the variable X01 is set.

USUAL FORMULA:

X01.M1 OR X01.M2 OR X01.M3 OR X01.M4 OR X01.M5 OR X01.M6 OR X01.M7 OR X01.M8

This query can be made much easier by the logical ORing of the overall status.

LOGICAL ORING:

X01.Status & 0xFF

The constant can be entered in hexadecimals, as described above:

0xFF corresponds to decimal 256; these are the first eight status bits (binary 11111111). If one of these bit is set to 1, the result of this bitwise ORing is 1 (true), otherwise it is 0 (false).

If, for example, all user status bits except the user status bit M7 should be queried, the binary statement for this would be: 10111111. Bit 7 is not of interest and is thus set to 0. This corresponds to 0xBF in hexadecimal. The expression for the formula is then: **X01.Status & 0xBF**.

Instead of ORing bitwise with a constant, the value can also be directly compared to a decimal number. If the comparison is wrong, the binary value is 0 (false) otherwise it is 1 (true).

Example:

You want to find out if the value is equal to the constant 202: The formula is:

X01.value = 202

If the value is equal to the constant 202, the result of the comparison is 1 (true) otherwise it is 0 (false).

Note: The bitwise ORing works with the OR character (|) in a similar manner to this example.

Comparison operators

Comparison operators are for the direct comparison of two numeric values. The result of this comparison is a binary value. "0" if the condition is not fulfilled and „1“ if the condition is fulfilled.

Operator	Description
<	Less
>	greater
<=	Less than or equal
>=	greater or equal
=	Equal
<>	unequal

To the left and to the right of the comparison operator, there has to be a (total) value or a (total) status, single bits cannot be used with these comparison operators.

There can also be a constant to the right of the comparison operator.

These constants are entered as hexadecimal values or decimal values in the combined element.

Hexadecimal figures are automatically converted to decimal values by clicking on **OK** (for example, 0x64 is in decimal figures 100).



Example

X01.value >= X02.value

The result is 1, if the value of X01 is higher than or equal to the value of X02

X01.value = 0x64

The result is 1, if the value of X01 is exactly equal to the numeric value 100 (= hex 0x64)

(X01.value = 0x64) OR (X01.value = 0x65)

The result is 1, if the value of X01 is exactly equal to the numeric value 100 or 101 (= hex 0x64 and hex 0x65)

Examples for formulas

SIMPLE LOGICAL AND LINKING BETWEEN TWO BIT VALUES



Example

Formula: X01.03 AND X02.03

This formula has the status TRUE, if both **bit 3** of variable 1 and **bit 3** of variable 2 both have the value 1.

COMPARISON OF AN ANALOGUE VALUE OR STATUS OF A VARIABLE



Example

(X01.Value > X02.Value)

COMPARE ANALOG VALUES WITH EACH OTHER ON A LOGICAL BASIS



Example

(X01.Value > X02.Value) AND (X01.Value = X02.Value)

COMPARE WITH VALUE BITS AND STATUS BITS



Example

(X01.Value > X02.Value) AND (X01.Value = X02.Value) OR (X01.03 = X02.03)

COMPARE A VALUE WITH A DECIMAL OR HEXADECIMAL VALUE



Example

Formula: *(X01.Value = 111)*

Formula: *(X01.Value = 0x6F)*

If a hexadecimal values is used, this is later transferred to decimal by clicking on **OK**. If a decimal value is entered and confirmed, the value continues to be displayed as a decimal value after reopening.



Info

It is not possible to use a comma or a period when entering values.

4.2.4 Create menu

Command Processing can also be activated via a context menu. Context menus are created in the Editor using node **Menus** and are defined in the properties of the element they concern.

Generally there are three types of menu entries:

Parameters	Description
Action type	<p>Sets out which type of action is to be carried out via the corresponding menu item in Runtime. Not all action types are available in the main menu, some are only available via the context menu.</p> <ul style="list-style-type: none"> ▶ Acknowledge alarm (context menu only) ▶ Command input (context menu only) ▶ Acknowledge flashing (context menu only) ▶ Function ▶ Help ▶ No action ▶ Send value to hardware ▶ VBA macro (context menu only)
Submenu	Opens a sub-menu in Runtime.
Separator	A horizontal line divides menu entries.

Underline text: Entering a & causes the following characters to be displayed as underlined.

Plan entries

To configure a menu item in the main menu or context menu:

1. Activate the corresponding menu cell
2. In properties, select:
 - **Action type:** depending on menu type
see also: Main menu action types and Context menu action types
 - **Menu ID:** Entry ID
Note: There are pre-defined types with a fixed ID available in the command input at Command input
 - **Text:** clear labeling of the menu cells



Attention

*The name of the **Text** property must be unique. If two names the same are issued, further menu items with the same name are not displayed.*

You can find details on the definition on context menus for command processing in chapter menusCommand Processing.

4.2.5 Create Runtime files - start the Runtime

When creating Runtime files for the command groups, a check for engineering errors and correct replacement is performed.

For every variable which has a command group allocated to it, a specific version of the command group for zenon is created. This version only contains **these** actions which can be triggered over this variable.



Example

The command group for the command variable only has actions on this command variable anymore. Except for the action 'Lock'. This action is also available for the command variable.



Information

The compiling of the command processing must also be triggered after changes to the variables.

Replace links

In order to increase the reusability of the command group, there is the possibility to replace the variable references. Replacement is possible for the response, command and condition variable.

During the replacement, the placeholder, '*', is automatically replaced by the name of the variables that are assigned to the command processing.



Example

Let's assume that we have the variables xyz_RM, abcRM and bool_RM.

Our mask is '*_RM'.

Variable name	Replacement text	Result	Comment
xyz_RM	xyz	xyz_RM	Variable exists, assignment is possible
abcRM	<empty>	_RM	The mask is not correct, because '_' is missing. Variable does not exist.
bool_RM	bool	bool_RM	Variable exists, assignment is possible



Information

If it is detected that the variable for which the command group is to be compiled corresponds to the response variable. The replacement text is determined by the response variable. Otherwise the replacement text is defined by the action variable or the first action appropriate to the variable.

If the replacement text was determined correctly, the placeholder '' is replaced by this.*

The following points should therefore be taken into account with the labeling:

- ▶ The names of the variables and the mask should be selected in such a way that these can be clearly assigned.
- ▶ The names of the variables that are used for the response, command and condition variables should be able to be created from the same replacement text.
- ▶ If the response variable is replaced but not the command variable, particular care should be taken to ensure that the command group that is created for the command variable also uses the expected response variable.
- ▶ Another test run for the command group with the response variable ensures that only actions whose action variable uses the same response variable with its compiled interlocking are contained. Actions that violate this rule create a warning and are removed.

Error while creating Runtime data

At the creation of the Runtime data for the command processing, an extensive validation is carried out concerning wrong engineering and not-available references.



Information

After an `Error` the object that caused the error is not available during runtime.

If the command group has an `Error`, no command group is assigned to the variable. Consequently, during the Runtime, all user operations are locked.

A Warning is generated when the project would cause a problem but runs error-free.

In the error messages, the following placeholders are used:

<VERNAME>	Placeholder is replaced in the error message by the name of the command group.
<VERRM>	Placeholder is replaced in the error message by the name of the response variable.
<AUFVAR>	Placeholder is replaced in the error message by the name of the variable to which the command group is assigned.
<ACTVAR>	Placeholder is replaced in the error message by the name of the variable of the action.
<Actionname>	Placeholder is replaced in the error message by the description of the action.
<VARNAME>	Placeholder is replaced by the variable in the visualization.

The following error messages can occur during the creation of the Runtime files:

Message text	Description
<VERNAME>: Interlocking PV <VERRM> does not exist!	Condition variable for general interlocking not available.
Variable '<AUFVAR>' uses not existing command processing!	Variable uses a non-existing command processing.
(<AUFVAR>) command processing '<VERNAME>' contains no actions!	Command groups without action are not considered by the Runtime. This message can also be a follow-up error.
(<AUFVAR>) response variable '<VARRM>' does not use the command group '<VERNAME>'	A response variable using another command group is used. The response variable always has to be linked with the interlocking, which uses it as response variable.
(<AUFVAR>) response variable '<VARRM>' for command processing '<VERNAME>' uses a driver without process linking!	The response variable must lie on a driver with process connection.
(<AUFVAR>) Command processing '<VERNAME>' contains no actions after compiling!	A command group without actions does not make sense.
(<AUFVAR>) response variable '<VARRM>' of command processing '<VERNAME>' not available!	The used response variable is not present or marked as deleted.
(<AUFVAR>) command processing '<VERNAME>' uses screen '<Bild GUID>'(<BILDNAME>) which is not of the type Power!	This message is a warning. If a user action becomes necessary during execution, it cannot be performed.
(<AUFVAR>) command processing '<VERNAME>' uses not available screen '<Bild GUID>'!	The screen assigned to the command group does not exist.
(<AUFVAR>) Replaced action variable '<ACTVAR>' for action '<Action name>' of command processing '<VERNAME>' not available!	The action variable, after a replacement, is not present or marked as deleted.
(<AUFVAR>) action '<actionname>' of the command '<VERNAME>' uses the not existing variable '<ACTVAR>'	The action uses a variable which is not present in the project or marked as deleted.
(<AUFVAR>) action variable '<ACTVAR>' for action '<Actionname>' of command processing '<VERNAME>' uses a driver without process connection!	The variable assigned to an action must not lie on an internal driver.
<VERNAME>(<AUFVAR>): Aktion '<Actionname>' already exists!	<p>The following actions may only be configured once per action variable and command group:</p> <ul style="list-style-type: none"> ▶ Switching command with the same command processing status. ▶ Correction with the same switching direction. ▶ Replacing with the same switching direction. ▶ Revision with the same switching direction.

	<ul style="list-style-type: none"> ▶ Pulse command ▶ Setpoint input ▶ Release ▶ Block ▶ Lock
<VERNAME>(<AUFVAR>): Action '<Actionname>': Pulse and switching command not possible for the same command variable!	Pulse and switching command must not be used together.
(<AUFVAR>) Action '<Actionname>' of the command processing '<VERNAME>' uses screen '<Bild GUID>(<Bildname>)' which is not of type Power!	This message is a warning. No user actions will be possible.
(<AUFVAR>) Action '<Actionname>' of the command processing '<VERNAME>' uses not existing screen '<Bild GUID>'	The action is assigned to a non-existing screen.
<VERNAME>(<AUFVAR>): Interlocking PV '<VARNAME>' does not exist!	Replaced condition variable does not exist.
(<AUFVAR>) Variable '<VARNAME>' of action interlocking condition of the command processing '<VERNAME>' does not exist!	Variable of the interlocking condition does not exist.
(<AUFVAR>) action variable '<VARNAME>' for action '<Actionname>' of command group '<VERNAME>' uses another command group!	The action variables used for a command group may only be connected to no command group or to the command group in which they are used.
(<AUFVAR>) command variable <ACTVAR> does not have a validly compiled interlocking! Action <Actionname> removed.	The action variable used in the action has no compiled command group. This message can also be a follow-up error.
(<AUFVAR>) command variable <ACTVAR> uses response variable <VARNAME>! Action <Actionname> removed.	<p>The compiled command group of the response variable contains actions with action variables which do not use the same response variable as <AUFVAR>.</p> <p>Note: There must not be any actions of response variables changing other response variables.</p>

4.3 Operating during Runtime

A watchdog timer is automatically carried out in the background if a user enters commands in Runtime.

4.3.1 Command procedure

This description for the procedure of a command with the command processing is applicable for the following command action types:

- ▶ Command (on page 78)
- ▶ Auto/Remote command (on page 79)
- ▶ Mandatory command (only in part) (on page 80)
- ▶ Setpoint input (on page 80)

The procedure of a command depends on the following parameters:

VARIABLE PROPERTY:

- ▶ Select before Operate (SBO) Not active:
In this case, no Select-command is executed by a command processing function.
Note: A driver (for example: DNP3_NG or IEC850) can nevertheless execute a Select automatically. This has no effect on the command processing however. The command processing will, in this case, react to an unsuccessful Select in the same way as an unsuccessful Operate/Execute .
- ▶ Select before Operate (SBO) active:
If Select before Operate is *active* for the action variable, a Select (`COT_act(6)` , `SE`) is always forwarded to the driver by the command processing. In doing so, there is a wait - regardless of the type of **runtime transfer** configured - until the complete command process has been completed.

Attention: The **SBO** variable property has corresponding effects on **runtime monitoring**.

If **SBO** is activated, the action buttons (and context menu entries) are deactivated in runtime monitoring for each configuration until `COT` contains the value `COT_accterm(10)`.

This also applies if "none" or "via response variable" are configured for runtime monitoring.

If no `COT_accterm` is received, only a **TIMEOUT** status bit is set.

 **Info**

If a variable is configured with an active **Select before Operate** and the driver does not support a **COT**, then no reaction to a sent Select will "come" from the PLC. Once the configured **Timeout** has expired, the command processing screen will inform you of the "**Timeout on SBO activation**" interlocking condition.

TYPE OF RUNTIME MONITORING:

- ▶ None
 - Without SBO: "fire & forget" - sends command and does not wait.
 - With SBO: there is a wait in the background, but no `TIMEOUT` status bit is set even if the **timeout** has expired.
- ▶ Via cause of transmission only (COT)
- ▶ Only via response variable (RV)
- ▶ Via RV and COT

PROPERTIES IN THE COMMAND PROCESSING ACTION:

- ▶ One-step:
After a successful Select, the command processing automatically transfers the Operate/Execute to the driver.
- ▶ Two-step:
A successful Select activates the buttons `Execute 2nd Step` and `Cancel` in the command processing screen:
 - If `Execute 2nd Step` is clicked on, the command processing transfers an Operate/Execute (`COT_act(6)`, no SE) to the driver.
 - If `Cancel` is clicked on, a Cancel (`COT_deact(8)`, SE) is forwarded to the driver via command processing.
- ▶ **Timeout:**
The configured value states how long is waited for a response from the PLC. Respectively:
 - After writing an Select command.
 - After writing an Execute command.
 - After writing an Cancel command.
- ▶ **Timeout can be canceled:**
Configuration states whether the user can cancel **runtime monitoring** after setting an Execute command.

Select before Operate

The **runtime monitoring** property only has an influence on the Execute command. The Select command is not influenced by **runtime monitoring**. The command processing action waits for as long as is configured in the **Timeout** property (**Options** property group in the command action) after a Select wartet has been sent.

In doing so, this can lead to the following dependencies:

- ▶ SPS does not react (on page 122)
- ▶ SPS reacts negatively (on page 123)
- ▶ SPS reacts positively (on page 123)
 - User does not send an Execute or Cancel (on page 124)

FOR CANCELING A SELECT - REGARDLESS OF THE TYPE OF RUNTIME MONITORING CONFIGURED - THE FOLLOWING IS APPLICABLE:

1. If **SBO** has been activated for the action variable, a **Select** can be canceled by the user for **two-stage** commands. After a successful Select (COT=7, SE), the command processing can send a Deactivation (COT=8, SE) if the **Cancel** button is clicked.
2. There is then a wait for a response from the PLC or a **Timeout**.
The action ends:
 - a) if the configured **timeout time** has expired or
 - b) if the PLC confirms the cancellation (COT=9, any SE or PN).
The receipt of a COT=9 discards the PROGRESS status bit.
3. The measurement time starts at the point when the user clicks the **Cancel** button. The time period that has expired - whilst the action waits for a Select - does not influence the time period in which the action waits for a confirmation of the cancellation.
4. The response variable does not receive a TIMEOUT status bit.
5. The same action always applies, regardless of the configuration of the **runtime monitoring** property.

Note: the **PN** bit - Positive(0)/Negative(1) - is reflected on the status bit **N_CONF** of the action variable.

SBO - no reaction from the PLC

The command processing first sends a Select to the driver. All buttons in the command processing window are grayed out while the action waits for a reaction from the PLC. Information is shown in the command processing screen once the **timeout** action has expired. This message is displayed in the field of the interlocking text: internal interlocking condition "**Timeout for SBO activation**". In this case, only the **Cancel** button is available. All other buttons of the command processing window are still grayed out.

Entries in the context menu are also not available.

Note: The response variable contains neither a `TIMEOUT` nor a `PROGRESS` status bit.

The same action applies for Select, regardless of which value is configured for **runtime monitoring**. The process is the same for two-step actions and one-step actions.

SBO - negative reaction from the PLC

The command processing triggers the driver to send a Select (`COT_act(6) + SE`). If a negative response is received by the PLC (`COT_actcon(7) + SE + PN=1`), waiting is no longer carried out.

- ▶ The `PROGRESS` bit is removed by clicking on the **Cancel** button. The response variable does not receive a `TIMEOUT` status bit in the process.
- ▶ If the action receives a negative response to Select, this information is displayed in the field of the interlocking test: internal interlocking condition "**SBO rejected**". In this case, only the **Cancel** button is available. All other buttons of the command processing window are still grayed out.

Entries in the context menu are also not available.

The same action applies to Select, regardless of which value is configured for **runtime monitoring**. The process is the same for two-step actions and one-step actions.

Note: the `PN` bit - Positive(0)/Negative(1) - is reflected on the status bit `N_CONF` of the action variable.

SBO - positive reaction from the PLC

If the PLC reacts positively to a Select (`COT_actcon(7) + SE`) (thus a confirmed Select), the action goes to the next step => execution

- ▶ The command processing ends the waiting for a Select.
- ▶ The response variable has its `PROGRESS` status bit set. However this is only if the current value of the response variable is different to the value of the command.

The following applies once the Select has been confirmed:

- a) One-step commands:
The command processing automatically sends an Operate/Execute (`COT_act(6)`, no `SE`) to the PLC.
- b) Two-stage actions:
The **Execute 2nd step** and **Cancel** in the command processing screen become active. If a

user confirms the **Execute 2nd step** button, the command processing sends an Operate/Execute to the PLC.

The **PROGRESS** bit is reset if the PLC confirms the Execute (`COT_actcon(7)`) or after expiry of the following **watchdog timer**.

SBO - positive reaction from the PLC but the user does not send an Execute or Cancel

If the user, after a successful Select, triggers neither an Execute nor a Cancel, there is a wait for user interaction with no time limitation. It can thus happen - provided the PLC supports **Select Timeout** - that this time expires. If this happens, the PLC sends a Select-Termination (`COT=10, SE, PN=1`). The command processing reacts to the Select-Termination received so that - if the user does in fact trigger an Execute - the command automatically sends another Select first.

Once the **Cancel** button in a command processing screen has been clicked on, a Select or Cancel is not sent to the PLC again.

Note: Is only relevant for two-step actions.

Watchdog timer

The cause of transmission (COT) is used to exchange information between zenon and the controller about whether a command is to be written or whether writing was successful. The action variable receives a `COT` corresponding to the level of the command. In the background, the command processing then checks to see if the response variable then changes its value and if the `COT` action variable changes according to the command.

Note: Value changes of the response variable will only be taken into account after `COT_act(6)`.



Information

`COTx` Status bits result in a value. This value can be evaluated in Runtime - just like all other status bits - using multi-numeric or multi-binary reaction matrices.

***Note:** `COT` is supported not only by IEC870, but also by some other Energy drivers - different versions thereof. Some drivers support `COT` although the protocol itself does not contain `COT` (e.g. IEC850, DNP3). You can find details in the corresponding driver documentation.*

RUNTIME MONITORING AND INTERLOCKING CONDITIONS OF THE ACTION:

Interlockings are checked during runtime monitoring:

1. For direct execution (= without Select).

2. When the Select is activated (= before Select).
 3. If no Select is pending (Select is OK but Execute has failed, because of a change in condition).
- ▶ The current command action shows messages about active interlockings. The user must unlock these. To do this, they must have the corresponding user rights. The interlocking must also be able to be unlocked. Otherwise this action can only be canceled.
 - ▶ If this action is canceled by a configured interlocking, no command is sent to the PLC.
 - ▶ If a Select has already been sent, cancellation is automatic. In this case, a Deactivation is sent.
 - ▶ The response variable does not receive a `TIMEOUT` bit in the process. If a `PROGRESS` bit is already set, this is reset.

WITH TWO-STEP CONFIGURATION OF THE ACTION:

The two-step action checks the interlocking and provides a message during the first step:

- ▶ **Direct execution - no SBO:**
Applicable for the moment when the first button is clicked.
- ▶ **Active SBO:**
Applicable for the moment when the confirmation for Select is received.

*By clicking on the button to unlock the interlocking, **Execute 2nd step** becomes available. If the button is clicked on, an Execute is sent to the PLC. If **Cancel** has been clicked on and a Select has already been sent, the action sends a Deactivation (`COT_deact (8) +SE`) to the PLC.*

WITH ONE-STEP CONFIGURATION OF THE ACTION:

*If there is an interlocking in the first stage, a message is displayed and there is a wait for a reaction from the user. If the user selected **On** or **Off**, the **Confirm** button will be active. The value is then sent to the PLC. If the user clicks on **Cancel** and a Select was carried out beforehand, the action sends a Deactivation (Cancel).*

*If there was no outstanding interlocking for the first stage before the conditions have changed and before the user has carried out a **Confirm** by clicking, the action will check the interlocking conditions again. If there are then still interlockings pending, a message is displayed and there is a wait for a cancellation from the user.*

Note: In this case, the interlocking cannot be unlocked by a user.



Information

Single-step actions have the same action in all scenarios.

RUNTIME MONITORING AND NEGATIVE RESPONSES FROM THE PLC:

- ▶ **Execute negative**
During runtime monitoring, the PLC can react negatively to an Execute/Operate (COT_actcon(7) + PN) . If the negative ration to an Execute was received, runtime monitoring is ended.
- ▶ **Execute Termination negative**
If Execution Termination (COT_actterm(10) + PN) is reported as negative by the PLC, runtime monitoring will no longer wait for a value change to the response variable and ends immediately.

The **PROGRESS** status bit is reset. The **TIMEOUT** status bit is not set for the response variable.

Note: the **PN** bit - Positive(0)/Negative(1) - is reflected on the status bit **N_CONF** of the action variable.

Cancellation of runtime monitoring

If the **Timeout can be cancelled** property has been activated for a command processing action, the user can cancel runtime monitoring with the "**Cancel**" button.

In doing so, the command processing sends the cancellation to the driver. The driver (e.g. IEC850) only forwards a Cancel Request to the PLC if the **Cancel Operate** property is active, then the user can also cancel using the **Cancel** button.

Note: Not all drivers support this cancellation of an Execute. You can find further information in the respective driver documentation.

Runtime monitoring via response variable only (RV)

Runtime monitoring via the response variable is the most-used type of runtime checking. It reacts to a change of the value of the response variable. The value that is expected for the response variable due to the command is defined in the **response status/switching direction** - on/off.

Negative responses from the PLC (COT_actcon(7) + PN) end runtime monitoring.

Note: Changes to the value of the response variable are only taken into account after COT_act(6) of the action variable. Prior value changes of the response variable - for example already after Select without an Operate/Execute having been sent - are ignored.

Runtime monitoring via cause of transmission only (COT)

With **runtime monitoring** via COT only, runtime monitoring does not react to the value of the response variable (RV) but only to the cause of transmission - the status bits COTx status bits of the action variable.

EXAMPLE WITHOUT SBO:

The process in detail:

1. The value and `COT_act (6)` are sent to the action variable.
2. The PROGRESS status bit is sent to the response variable.
3. If the controller receives the value `COT_act`, there is a wait for the subsequent values `COT_actcon(7)` and `COT_actterm(10)`.
4. End of the process:
 - a) The process is ended if `COT_actterm` has been received.
 - b) If, in the configured timeout time, no `COT_actterm` has been received, then:
 - the process is ended and
 - the TIMEOUT status bit of the response variable is activated.

Note: You configure this time in the **Timeout** property for the command action.
5. The PROGRESS status bit is reset.

Runtime monitoring via COT (cause of transmission) and RV (response variable)

With **runtime monitoring** via COT and RV, the runtime monitoring reacts to the value of the response variable (RV) and to the cause of transfer - the `COTx` status bits of the action variable.

EXAMPLE WITHOUT SBO:

The process in detail:

1. The value and `COT_act (6)` are sent to the action variable.
2. The PROGRESS status bit is sent to the response variable.
3. If the controller receives the value `COT_act`, there is a wait for the subsequent values `COT_actcon(7)` or `COT_actterm (10)`.
4. End of the process:
 - a) The process is ended if both conditions have been met:
 - `COT_actterm` was received
 - and**
 - The value of the response variables corresponds to the **switching direction (response status/ switching direction property)**.

It does not matter which of the two conditions is met first. As soon as both of them are fulfilled, the procedure will be terminated.

- a) If only one or none of the above conditions from item 4a is met within the configured **timeout**, then:
 - the process is ended and will be terminated and
 - the TIMEOUT status bit of the response variable is activated.
5. The PROGRESS status bit is reset.

4.3.2 Command Processing screen control elements

A command processing screen allows control in Runtime and an overview of the command processing. The command processing can be controlled via buttons.

CREATING A SCREEN OF THE TYPE COMMAND PROCESSING

To create a command processing screen:

1. Select, in the toolbar or in the context menu of the **Screens** node, the **New Screen** command
2. An standard empty screen is opened
3. Change the screen type in the detail view; to do this:
 - a) click on **Standard** in the **Screen type** column
 - b) Select **Command Processing** from the drop down list
4. Click in the screen.
5. Select the **Control elements** menu item in the menu bar
6. Click on **Add template** in the drop-down list
7. The standard elements are inserted
8. Select additional elements as required and insert them into the desired place on the screen
9. Create a screen switch function, in order to be able to call up the Command Processing in Runtime

COMMAND PROCESSING SCREEN TEMPLATE - STANDARD

Action/command	Interlockings
<div>Active action/command</div> <div>Active action Tvn: STATIC</div> <div>Desired switching direction</div> <div>Desired switching direction Tvn: STATIC</div>	<div>Active interlocking</div> <div>Interlocking text Typ: STATIC ID: 10034</div> <div>Unlock</div>

Select/execute

True

False

Confirm

Cancel

The following control elements are available in Runtime:

Control element	Description
Insert template	<p>Opens the dialog for selecting a template for the screen type.</p> <p>Templates are shipped together with zenon and can also be created by the user.</p> <p>Templates add pre-defined control elements to pre-defined locations in the screen. Elements that are not necessary can also be removed individually once they have been created. Additional elements are selected from the drop-down list and placed in the screen. Elements can be moved on the screen and arranged individually.</p>
Action/command	
Active action/command	Displays the pending action of the command group.
Switching direction	<p>The switching direction configured for the active action. The texts are documented with the "switching direction" setting.</p> <p>Depending on the active action, the following text is shown:</p> <p>Command, revision, correction, replace: Text from limit value, depending on switching direction.</p> <p>Status: On or Off</p> <p>Other: empty</p>
Interlockings	
Active interlocking	Interlocking text of the active interlocking.
Unlock	<p>If an unlockable interlocking is upcoming, it can be unlocked with this button.</p> <p>Note: This control is shown only when the screen is in the step 'Unlock'.</p> <p>The Control is locked when the upcoming interlocking is not unlockable.</p>
Select / execute	
On	Command button for switching command, to close a switch for example.
Off	Command button for switching command, to open a switch for example.
Confirm	<p>Confirms for the pending two-step action.</p> <p>A two-step switching command, for example, is only executed after clicking on this button.</p>

Cancel	<ul style="list-style-type: none">▶ Closes command processing screen. The pending action is not executed.▶ Cancel for pending, two-step action.▶ Cancellation of the execution of an action (depending on project configuration, for example: Operate in the case of SBO). <p>The button is grayed out if the screen is in 'Step 1'.</p>
---------------	--

Name	Control type		Default
Action buttons	Text	<p>Buttons, which can have an action assigned to them. By clicking in the screen, the assigned action is activated and the screen changes to the step "Release"</p> <p>The button is not shown when:</p> <ul style="list-style-type: none"> ▶ No action is assigned to the button in the current command group. ▶ The variable, with which the screen was loaded, is the command variable, and the action assigned to the button does not use the command variable as action variable. However, if the action 'Lock' was assigned to the button, it is visible. <p>The button is shown as locked when:</p> <ul style="list-style-type: none"> ▶ The screen is not in 'Step 1'. ▶ The response variable has set one of the status bits I_KENNUNG(18), OFF(20) or NICHT_AKTUELL(29) and writes the assigned action to the command variable. ▶ The response variable has the status REVISION(9) active and the assigned action writes to the command variable. ▶ The response variable has the status REVISION(9) active and the assigned action is 'Correct'. ▶ The assigned action is 'Release' and the response variable does not have the status Alternativevalue(27) active. ▶ The assigned action is 'Correct' and the value of the response variable matches the switching direction. ▶ The assigned action is 'Replace' and the value of the response variable matches the switching direction. ▶ The response variable has the status REVISION(9) active and the assigned action is 'Replace'. ▶ The assigned action is 'Revision' and the value of the response variable 	Action1 Action2

		matches the switching direction.	
--	--	----------------------------------	--

RV TTA	Text	Name of the response variable	X
RV identification	Text	Name of the response variable	X
Action variable unit	Text	Unit of the current action variable.	X
Action variable set status	List	<p>Defines the status to be set for the action 'Status default' for the switching direction 'None'. The statuses are set to the current status and updated when changes occur.</p> <p>Is locked when the active action is not 'Set status'.</p>	
Switching direction	Text	<p>The switching direction configured for the active action. The texts are documented with the setting 'Switching direction'.</p> <p>Depending on the active action, the following text is shown:</p> <p>Command, revision, correction, replace: Text from limit value, depending on switching direction.</p> <p>Status: On or Off</p> <p>Other: empty</p>	X
Execute Step 2	Button	<p>Delivers the actions to execution.</p> <p>This control is visible only when the screen is in 'Step 2'.</p> <p>The Control is locked when:</p> <ul style="list-style-type: none"> ▶ Two handed operation was configured and the key 'Ctrl' is not pressed. ▶ The status REVISION(9) of the response variable is set and the assigned action is 'Command', 'Set value', 'Replace' or 'Correct'. ▶ The button was already clicked. 	X
Action variable minimum	Numeric	<p>Minimum value of the action variable.</p> <p>Not visible if the action variable is of data type 'String'.</p>	
Action variable maximum	Numeric	<p>Minimum value of the action variable.</p> <p>Not visible if the action variable is of data type 'String'.</p>	
Scrollbars	Numeric	Setpoint input with scroll bar Sets the value in the control 'Set value' or is set by this value.	

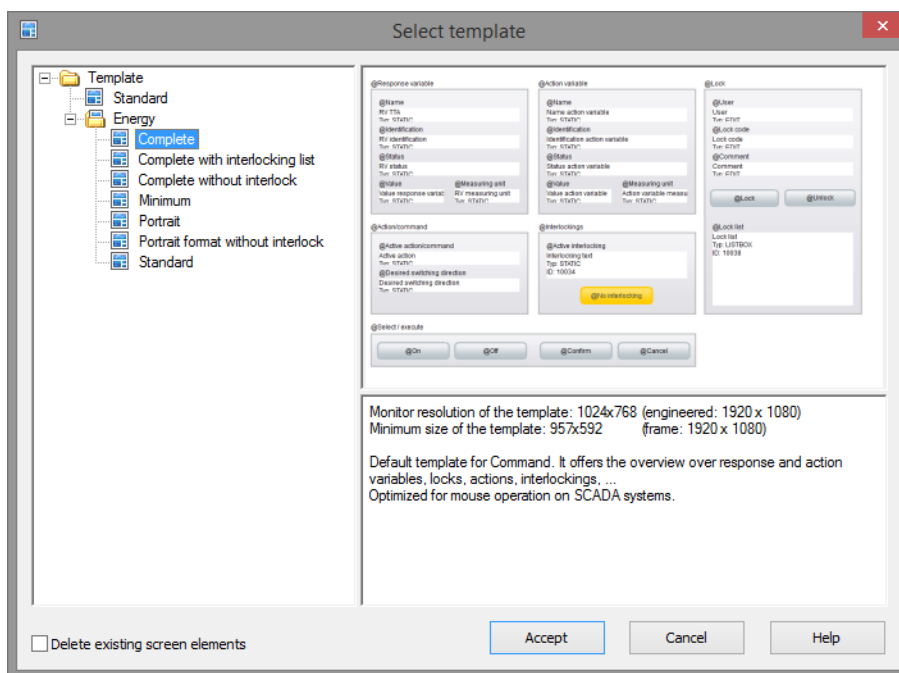
		<p>Not visible if the action variable is of data type 'String'.</p> <p>The Control is locked when:</p> <ul style="list-style-type: none"> - No action is active. - The screen is not in 'Step 1'. 	
Set value	Numerical, Text	<p>Allows the input of the set value.</p> <p>By clicking the Control, it is switched to edit mode and the setpoint input is possible. The edit mode can be left again with "Enter".</p> <p>The new value is set only after clicking the control 'Execute'.</p> <p>The desired value for the action 'Set value' is provided with this control.</p> <p>The Control is locked when:</p> <ul style="list-style-type: none"> - The status REVISION(9) of the response variable is set. - No action is active. - The screen is not in 'Step 1'. 	
RV value	Text	Value of the response variable	X
RV status	Text	Contains the status of the response variable in the short form.	X
RV unit	Text	Unit of the response variable	X
Interlocking text	Text	<p>Text of the upcoming interlocking.</p> <p>Text is online language switchable</p>	X
No interlocking		<p>If an unlockable interlocking is upcoming, it can be unlocked with this button.</p> <p>Note: This control is shown only when the screen is in the step 'Unlock'.</p> <p>The Control is locked when the upcoming interlocking is not unlockable.</p>	X
Exit	Button	<p>Closes the screen without action execution.</p> <p>The button is only visible in a modal screen.</p> <p>This button is important for modal screens, because it is required to leave the screen in case of an error!</p>	
Cancel	Button	Aborts the execution of the Command Processing and returns to 'Step 1'.	X

		The button is locked when the screen is in 'Step 1'.	
Lock list	List	<p>Contains the locks that were activated at the response variable.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p> <p>Text is online language switchable</p>	
User identification	Input field	<p>For entering the user identification for the lock.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p>	
Lock code	Input field	<p>For entering the user-specific lock code.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p>	
Execute lock	Button	<p>Activates a lock for the user entered in the Control 'User identification'.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p> <p>This user action is logged in the CEL, if not suppressed by the engineering.</p>	
Unlock	Button	<p>Removes the lock by the user entered in the user identification.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p> <p>This user action is logged in the CEL, if not suppressed by the engineering.</p>	
Execute	Button	<p>Takes over the value of the Control 'Set value' or 'Set status'</p> <p>This Control is visible only when the screen is in 'Step 1'.</p> <p>The Control is locked additionally to the general lock, when:</p> <ul style="list-style-type: none"> ▶ The active action is not 'Set status', 'Set value' or 'Correct set value'. ▶ The value in the control 'Set value' for the action variable is invalid. 	
Comment	Input field	Comment about the lock.	
Action variable Status	Text	Status of the active action variable in short form.	X

Action variable Name	Text	Name of the active action variable.	X
Action variable Identification	Text	Identification of the active action variable.	X
Action variable value	Text	Value of the active action variable.	X
Active action	Text	Name of the active action.	X

Template

Several pre-defined templates are available for **Command Processing** screens.



Template	Description
List field templates (left)	Displays all pre-defined and user-defined template.
Preview and description (right)	Shows preview and description of the selected template.
Standard	<p>Compact display of the command input with visualization of:</p> <ul style="list-style-type: none"> ▶ Actions ▶ Interlockings ▶ Buttons for selection and execution

ENERGY

Template	Description
Portrait Format	<p>Display of command input in portrait format, optimized for placing next to an overview screen:</p> <ul style="list-style-type: none"> ▶ Actions ▶ Interlockings ▶ Buttons for selection and execution ▶ Lock
Portrait format without interlock	<p>Simplified display of command input in portrait format:</p> <ul style="list-style-type: none"> ▶ Actions ▶ Interlockings ▶ Buttons for selection and execution
Complete	<p>Enhanced display of the command input with visualization of:</p> <ul style="list-style-type: none"> ▶ Response variables ▶ Action variables ▶ Lock ▶ Actions ▶ Interlockings ▶ Buttons for selection and execution
Complete with interlocking list	<p>Enhanced display of command input including all interlockings:</p> <ul style="list-style-type: none"> ▶ Response variables ▶ Action variables

	<ul style="list-style-type: none"> ▶ Interlocking (list) ▶ Actions ▶ Buttons for selection and execution
Complete without interlock	<p>Enhanced display of command input. With interlockings, only the currently-pending interlocking is visualized:</p> <ul style="list-style-type: none"> ▶ Response variables ▶ Action variables ▶ Actions ▶ Interlockings (Text) ▶ Buttons for selection and execution
Minimum	Only contains visualization of buttons for selection and execution (on page 143).

CLOSE DIALOG

Parameters	Description
Delete existing screen elements	<p>Behavior when applying the template for configuration in the Editor.</p> <p>Active: Already existing elements in the screen are deleted when taking over the template.</p> <p>Default: Not active.</p>
Apply	Inserts the elements of the selected template in the screen and closes the dialog.
Cancel	Closes the dialog without inserting elements.
Help	Opens online help.

Control elements - complete overview

XX

Parameters	Description

The xxx control element group takes on a special status:

Note of where more detailed explanations scan be found.

Control elements - group: Action/command

Action/command

Active action/command

Active action

T_{un}: STATIC

Desired switching direction

Desired switching direction

T_{un}: STATIC

Control element	Description
Active action/command	Type of active command action (Action type). Examples: Switching command, Check response value).
Switching direction	Value for the Return state/switching direction property configured for the active action. Depending on the active action, the following text is shown: <ul style="list-style-type: none"> ▶ Command ▶ Revision ▶ Manual correction ▶ Replace Text from limit value, depending on switching direction. ▶ Status: On or Off ▶ Other: empty

Control elements - group: Action variable

Action variable

Name	
Name action variable	
Tvn: STATIC	
Identification	
Identification action variable	
Tvn: STATIC	
Status	
Status action variable	
Tvn: STATIC	
Value	Measuring unit
Value action variable	Action variable meas
Tvn: STATIC	Tvn: STATIC

Control element	Description
Name	Name of the action variable.
Identification	Identification of the action variable.
Status	<p>Contains the short description of the status bits for the action variable.</p> <p>Example:</p> <ul style="list-style-type: none"> ▶ Bits for COT ▶ Status SE_870 during Select ▶ Status PN bit in the event of a negative response from the PLC <p>Note: The status bits contain the response variable for the "status input" action.</p>
Value	<p>Current value of the action variable or input field for setpoint input command processing action.</p> <p>Note: This value changes during the course of the action from an existing to a current value. The display of the value is only refreshed with COT=7 (COT_actcon) or WR-SUC .</p> <ul style="list-style-type: none"> ▶ The following is applicable for a configured setpoint input: The value to be set for the 'Set value' action is stipulated by this control element. By clicking the control element, it is switched to edit mode and the setpoint input is possible. Edit mode is left again by pressing Enter. However the new value is only set once the "Execute" control element has been clicked on. <p>The control element is locked when:</p> <ul style="list-style-type: none"> ▶ The status REVISION(9) of the response variable is set. ▶ No action is active. ▶ The screen is not in 'Step 1'.
Measuring unit	Measuring unit of the action variable

Control elements - group: Response variable

Response variable

Name	
RV TTA	
Tvn: STATIC	
Identification	
RV identification	
Tvn: STATIC	
Status	
RV status	
Tvn: STATIC	
Value	Measuring unit
Value response varia	RV measuring unit
Tvn: STATIC	Tvn: STATIC

Control element	Description
Name	Name of the response variable
Identification	Name of the response variable
Status	Contains the short description of the status bits for the response variable.
Value	Current value of the response variable
Measuring unit	Measuring unit of the response variable

Control elements - group: Select / execute

Select / execute

On	Off	Confirm	Cancel
----	-----	---------	--------

Control element	Description
On	First-step command button, to close a switch for example. Note: Only visible in Step 1.
Off	First-step command button, to open a switch for example. Note: Only visible in Step 1.
Confirm	Second-step command button. Confirms the process for first-step command action. Note: Only visible in Step 2.
Cancel	Second-stage command button. Aborts the execution of the command processing and returns to 'Step 1'. Note: Only visible in Step 2.
Close	Closes the command window in zenon Runtime

Control elements - group: Lock

Lock

User
User
Tvn: FNIT

Lock code
Lock code
Tvn: FNIT

Comment
Comment
Tvn: FNIT

Lock Unlock

Lock list
Lock list
Typ: LISTBOX
ID: 10038

Control elements from the **Lock** group are locked if no "Lock" action is configured in the command group.

Control element	Description
Users	For entering the user identification for the lock.
Lock code	For entering the user-specific lock code.
Comment	Optional text that can be entered by the user for the lock.
Lock	<p>Activates a lock by the user entered in the User control element.</p> <p>Note: This user action is logged in the CEL, if not suppressed by the engineering.</p>
Unlock	<p>Removes the lock that has been set up by the user entered in the User control element.</p> <p>This guarantees that only people's own locks can be removed.</p> <p>Note: This user action is logged in the CEL, if not suppressed by the engineering.</p>
Lock list	<p>List of active locks:</p> <ul style="list-style-type: none"> ▶ User Name of the user who has activated the lock. ▶ Locking time Time stamp of the interlocking ▶ Note Text for the interlocking.

Control elements - group: Interlockings

Interlockings

Active interlocking

Interlocking text

Typ: STATIC

ID: 10034

No interlocking

Control element	Description
Active interlocking	The active interlocking (on page 100) according to the configuration or texts from ALC - topological interlocking (on page 32).
Unlock	<p>This button unlocks an active, unlockable interlocking.</p> <p>Note: This control element is shown only when the screen is in the step 'Unlock'.</p> <p>The Control element is locked when the upcoming interlocking is not unlockable.</p>

Interlocking - list

Command Processing screen - Standard

XX

Action/command

Active action/command

Active action

Typ: STATIC

Desired switching direction

Desired switching direction

Typ: STATIC

Interlockings

Active interlocking

Interlocking text

Typ: STATIC

ID: 10034

Unlock

Select/execute

True False Confirm Cancel

ACTION/COMMAND

Control element	Description
Active action/command	Displays the pending action of the command group.
Switching direction	<p>The switching direction configured for the active action. The texts are documented with the "switching direction" setting.</p> <p>Depending on the active action, the following text is shown:</p> <ul style="list-style-type: none"> ▶ Command, revision, correction, replace: Text from limit value, depending on switching direction. ▶ Status: On or Off ▶ Other: empty

Interlockings	
Active interlocking	Interlocking text of the active interlocking.
Unlock	<p>If an unlockable interlocking is upcoming, it can be unlocked with this button.</p> <p>Note: This control is shown only when the screen is in the step 'Unlock'.</p> <p>The Control is locked when the upcoming interlocking is not unlockable.</p>

Select / execute	
On	Command button for switching command, to close a switch for example.
Off	Command button for switching command, to open a switch for example.
Confirm	<p>Confirms for the pending two-step action.</p> <p>A two-step switching command, for example, is only executed after clicking on this button.</p>
Cancel	<ul style="list-style-type: none"> ▶ Closes the command processing screen. The pending action is not executed. ▶ Cancel for pending, two-step action. ▶ Cancellation of the execution of an action (depending on project configuration, for example: Operate in the case of SBO). <p>The button is grayed out if the screen is in 'Step 1'.</p>

Name	Control type		Default
Action buttons	Text	<p>Buttons, which can have an action assigned to them. By clicking in the screen, the assigned action is activated and the screen changes to the step "Release"</p> <p>The button is not shown when:</p> <ul style="list-style-type: none"> ▶ No action is assigned to the button in the current command group. ▶ The variable, with which the screen was loaded, is the command variable, and the action assigned to the button does not use the command variable as action variable. However, if the action 'Lock' was assigned to the button, it is visible. <p>The button is shown as locked when:</p> <ul style="list-style-type: none"> ▶ The screen is not in 'Step 1'. ▶ The response variable has set one of the status bits I_KENNUNG(18), OFF(20) or NICHT_AKTUELL(29) and writes the assigned action to the command variable. ▶ The response variable has the status REVISION(9) active and the assigned action writes to the command variable. ▶ The response variable has the status REVISION(9) active and the assigned action is 'Correct'. ▶ The assigned action is 'Release' and the response variable does not have the status Alternativevalue(27) active. ▶ The assigned action is 'Correct' and the value of the response variable matches the switching direction. ▶ The assigned action is 'Replace' and the value of the response variable matches the switching direction. ▶ The response variable has the status REVISION(9) active and the assigned action is 'Replace'. ▶ The assigned action is 'Revision' and the value of the response variable 	<p>Action1</p> <p>Action2</p>

		matches the switching direction.	
--	--	----------------------------------	--

RV TTA	Text	Name of the response variable	X
RV identification	Text	Name of the response variable	X
Action variable unit	Text	Unit of the current action variable.	X
Action variable set status	List	Defines the status to be set for the action 'Status default' for the switching direction 'None'. The statuses are set to the current status and updated when changes occur. Is locked when the active action is not 'Set status'.	
Switching direction	Text	The switching direction configured for the active action. The texts are documented with the setting 'Switching direction'. Depending on the active action, the following text is shown: Command, revision, correction, replace: Text from limit value, depending on switching direction. Status: On or Off Other: empty	X
Execute Step 2	Button	Delivers the actions to execution. This control element is visible only when the screen is in 'Step 2'. The control element is locked when: <ul style="list-style-type: none"> ▶ Two handed operation was configured and the key 'Ctrl' is not pressed. ▶ The status REVISION(9) of the response variable is set and the assigned action is 'Command', 'Set value', 'Replace' or 'Correct'. ▶ The button was already clicked. 	X
Action variable minimum	Numeric	Minimum value of the action variable. Not visible if the action variable is of data type 'String'.	
Action variable maximum	Numeric	Minimum value of the action variable. Not visible if the action variable is of data type 'String'.	
Scrollbars	Numeric	Setpoint input with scroll bar Sets the value	

		<p>in the control element 'Set value' or is set by this value.</p> <p>Not visible if the action variable is of data type 'String'.</p> <p>The control element is locked when:</p> <ul style="list-style-type: none"> - No action is active. - The screen is not in 'Step 1'. 	
Set value	Numerical, Text	<p>Allows the input of the set value.</p> <p>By clicking the control element, it is switched to edit mode and the setpoint input is possible. The edit mode can be left again with "Enter".</p> <p>The new value is set only after clicking the control element 'Execute'.</p> <p>The desired value for the action 'Set value' is provided with this control element.</p> <p>The control element is locked when:</p> <ul style="list-style-type: none"> - The status REVISION(9) of the response variable is set. - No action is active. - The screen is not in 'Step 1'. 	
RV value	Text	Value of the response variable	X
RV status	Text	Contains the status of the response variable in the short form.	X
RV unit	Text	Unit of the response variable	X
Interlocking text	Text	<p>Text of the upcoming interlocking.</p> <p>Text is online language switchable</p>	X
No interlocking		<p>If an unlockable interlocking is upcoming, it can be unlocked with this button.</p> <p>Note: This control element is shown only when the screen is in the step 'Unlock'.</p> <p>The control element is locked when the upcoming interlocking is not unlockable.</p>	X
Exit	Button	<p>Closes the screen without action execution.</p> <p>The button is only visible in a modal screen.</p> <p>This button is important for modal screens,</p>	

		because it is required to leave the screen in case of an error!	
Cancel	Button	<p>Aborts the execution of the Command Processing and returns to 'Step 1'.</p> <p>The button is locked when the screen is in 'Step 1'.</p>	X
Lock list	List	<p>Contains the locks that were activated at the response variable.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p> <p>Text is online language switchable</p>	
User identification	Input field	<p>For entering the user identification for the lock.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p>	
Lock code	Input field	<p>For entering the user-specific lock code.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p>	
Execute lock	Button	<p>Activates a lock for the user entered in the control element 'User identification'.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p> <p>This user action is logged in the CEL, if not suppressed by the engineering.</p>	
Unlock	Button	<p>Removes the lock by the user entered in the user identification.</p> <p>Is locked when no action 'Lock' was configured for the command group.</p> <p>This user action is logged in the CEL, if not suppressed by the engineering.</p>	
Execute	Button	<p>Takes over the value of the control element 'Set value' or 'Set status'</p> <p>This control element is visible only when the screen is in 'Step 1'.</p> <p>The control element is locked additionally to the general lock, when:</p> <ul style="list-style-type: none"> ▶ The active action is not 'Set status', 'Set value' or 'Correct set value'. ▶ The value in the control element 'Set value' for the action variable is invalid. 	

Comment	Input field	Comment about the lock.	
Action variable Status	Text	Status of the active action variable in short form.	X
Action variable Name	Text	Name of the active action variable.	X
Action variable Identification	Text	Identification of the active action variable.	X
Action variable value	Text	Value of the active action variable.	X
Active action	Text	Name of the active action.	X

Command Processing screen - complete

A command processing screen allows control in Runtime and an overview of the command processing. The command processing can be controlled via buttons.

CREATING A SCREEN OF THE TYPE COMMAND PROCESSING

To create a command processing screen:

1. Select, in the toolbar or in the context menu of the **Screens** node, the **New Screen** command
2. An standard empty screen is opened
3. Change the screen type in the detail view; to do this:
 - a) click on **Standard** in the **Screen type** column
 - b) Select **Command Processing** from the drop down list
4. Click in the screen.
5. Select the **Control elements** menu item in the menu bar
6. Click on **Add template** in the drop-down list
7. Select, as a template, **Complete** in the *Energy* template folder
8. The standard elements are inserted
9. Select additional elements as required and insert them into the desired place on the screen

10. Create a screen switch function, in order to be able to call up the Command Processing in Runtime

Response variable	Action variable	Lock
Name RV TTA Tvp: STATIC Identification RV identification Tvp: STATIC Status RV status Tvp: STATIC Value Measuring unit Value of response vari. RV measuring unit Tvp: STATIC Tvp: STATIC	Name Action variable name Tvp: STATIC Identification Action variable identification Tvp: STATIC Status Action variable status Tvp: STATIC Value Measuring unit Action variable value Action variable measur Tvp: STATIC Tvp: STATIC	User User Tvp: EDIT Lock code Lock code Tvp: EDIT Comment Comment Tvp: EDIT <input type="button" value="Lock"/> <input type="button" value="Unlock"/>
Action/command Active action/command Active action Tvp: STATIC Desired switching direction Desired switching direction Tvp: STATIC	Interlockings Active interlocking Interlocking text Typ: STATIC ID: 10034 <input type="button" value="Unlock"/>	Lock list Lock list Typ: LISTBOX ID: 10038
Select / execute <input type="button" value="True"/> <input type="button" value="False"/> <input type="button" value="Confirm"/> <input type="button" value="Cancel"/>		

Parameters	Description
Response variable	
Name	Name of the response variable
Identification	Name of the response variable
Status	Contains the short description of the status bits for the response variable.
Value	Current value of the response variable
Measuring unit	Measuring unit of the response variable
Action variable	
Name	Name of the action variable
Identification	Identification of the action variable
Status	<p>Contains the short description of the status bits for the action variable.</p> <p>Example:</p> <ul style="list-style-type: none"> ▶ Bits for COT ▶ Status SE_870 during Select ▶ Status PN bit in the event of a negative response from the PLC <p>Note: The status bits contain the response variable for the "status input" action.</p>
Value	<p>Current value of the action variable or input field for setpoint input command processing action.</p> <p>Note: This value changes during the course of the action from an existing to a current value. The display of the value is only refreshed with COT=7 (COT_actcon) or WR-SUC .</p> <ul style="list-style-type: none"> ▶ The following is applicable for a configured setpoint input: The value to be set for the 'Set value' action is stipulated by this control. By clicking the Control, it is switched to edit mode and the setpoint input is possible. It is possible to leave the editing mode again by pressing the Enter key. However the new value is only set when the "Execute" control is clicked on. The control is blocked if: <ul style="list-style-type: none"> - The response variable has set the status REVISION(9). - No action is active. - The screen is not in the "Step 1" stage.

Measuring unit	Measuring unit of the action variable
Lock	Controls from the Lock group are locked if no "Lock" action is configured in the command group.
Users	For entering the user identification for the lock.
Lock code	For entering the user-specific lock code.
Comment	Optional text that can be entered by the user for the lock.
Lock	Activates a lock by the user entered in the "User" control. Note: This user action is logged in the CEL, if not suppressed by the engineering.
Unlock	Removes the lock that has been set up by the user entered in the "User" control. This guarantees that only people's own locks can be removed. Note: This user action is logged in the CEL, if not suppressed by the engineering.
Lock list	List of active locks: <ul style="list-style-type: none"> ▶ User Name of the user who has activated the lock. ▶ Locking time Time stamp of the interlocking ▶ Note Text for the interlocking.
Action/command	
Active action/command	Type of active command processing action such as dual command, for example.
Switching direction	The switching direction configured for the active action. The texts are documented with the setting 'Switching direction'. Depending on the active action, the following text is shown: Command, revision, correction, replace: Text from limit value, depending on switching direction. Status: On or Off Other: empty
Interlockings	
Active interlocking	The active interlocking (on page 100) according to the configuration or texts from ALC - topological interlocking (on page 32).
Unlock	This button unlocks an active, unlockable interlocking.

	<p>Note: This control is shown only when the screen is in the step 'Unlock'.</p> <p>The Control is locked when the upcoming interlocking is not unlockable.</p>
Select / execute	
On	<p>First-step command button, to close a switch for example.</p> <p>Note: Only visible in Step 1.</p>
Off	<p>First-step command button, to open a switch for example.</p> <p>Note: Only visible in Step 1.</p>
Confirm	<p>Second-step command button.</p> <p>Note: Only visible in Step 2.</p>
Cancel	<p>Second-stage command button.</p> <p>Aborts the execution of the command processing and returns to 'Step 1'.</p> <p>The button is grayed out if the screen is in 'Step 1'.</p>
Close	Closes the command processing screen.

ADDITIONAL CONTROL ELEMENTS

Actions	
Action 1 ... Action 20	<p>Buttons for desired actions of the 1st step.</p> <p>The command action is configured in the Command processing field property group in the Action button property window.</p> <p>Note: If the action is already assigned to a button, it is grayed out in the list.</p>
Information control element	
Minimum	Display of the minimum value of the set value.
Maximum	► Display of the maximum value of the set value.
Interlocking list	<p>Overview box of the active interlockings:</p> <ul style="list-style-type: none"> ► Text Text of the interlocking as configured ► Active (yes/no) ► Unlockable (yes/no) ► Unlocked (yes/no)

	► Type
Unlock all	Unlocks all active and unlockable interlockings

Blocked or locked elements

GENERAL LOCK OF THE CONTROLS

Some requirements must be met in order to unlock the controls in the screen. Since these requirements usually concern several controls, they are not listed with the control each time, but they are documented here.

1. All Controls except **Exit** are locked when:
 - the screen is not the owner of the active NET SEL
 - No command processing was configured for the add-on variable
 - the response variable does not exist
 - The response variable has not received a value yet
 - the INVALID bit status is active for the selected variable
 - an action for the action variable is running
 - Runtime monitoring for the action variable is running
 - **Note:** The **Cancel** button can be active here - regardless of the configuration.
 - There is a wait for the SBO confirmation from the Select
 - the data of the lock are being transmitted
 - the data of the lock are invalid
 - the currently-registered user does not have the necessary authorization levels
1. All Controls except **Exit** and the controls for the lock are locked when:
 - one of the locking conditions of point 1 apply
 - The response variable was locked against command processing
 - the status bit **S_MERKER_1(0)**, i.e. the command lock, of the response variable was set

COMMAND PROCESSING

- Action buttons:
 - Action buttons are locked if
 - The authorization level of the user who is logged in does not allow execution.

- The variable is locked.
 - There is already a command being executed
- Unlocking:
Unlocking is only possible if the user does not have the necessary authorization levels for this.
- Context menu:
Menu items that are assigned to a command processing action can only be selected if the registered user has the necessary authorizations.

LOADING A SCREEN WITH INITIAL STEP 'LOCK'

If the screen is loaded with the initial step 'Lock', all but the following controls are hidden in the screen:

- RV TTA
- RV identification
- RV value
- RV status
- RV measuring unit
- Lock code
- User identification
- Execute lock

4.3.3 Reload project online

If online reloading is instigated, the following effects must be expected:

If runtime monitoring, edge generation or SBO is active, the reloading is delayed until this has ended.

An opened command processing screen is closed and the process is started again after reloading depending on the current step:

Step before reloading	Action after reloading
Step 1	Screen is called up again for Step 1
Interlocking or Step 2	Unlocking step is activated. The interlocking is executed again.
Lock	Lock is activated again

Before it is called up again, the add-on variable, response variable, command group and action are determined again. The control elements are locked if one of the objects is no longer present.

If the command group was removed or replaced for the add-on variable, the screen with locked control elements is called up. The screen must then be called up again or the command processing must be executed again.

If the command group was removed or replaced for the response variable, all command processing locks are removed by the variables.

If a user who has activated a command lock no longer exists, the lock is removed. The status bit S_MERKER_1(0) is updated accordingly.

4.3.4 Logging in the CEL

In the CEL, the following user actions are logged in addition to the switching actions.

Parameters	Description
Unlock	The unlocking of an active interlocking is noted in the CEL.
Unlock all	A corresponding CEL entry is created for each unlocked interlocking.
Execute action	If the "Suppress CEL entry" action setting is not active, the execution of the action is logged in the CEL.

4.3.5 Lock return variable

For a locked response variable, the statusbit S_MERKER_1(0) is set. The lock can be activated or deactivated by entering the username and the lock code defined during the user definition. One lock can be activated per user. The active locks are remanent and are also considered after a system restart.



Information

The locks are automatically synchronized in the network; therefore, they can also be used in redundant operation.

4.3.6 Server change in redundant operation

If the Primary Server changes, the Select object is lost and the command processing must be executed again. The same applies for the SBO.

4.3.7 Exit Runtime

As long as there are still active actions in the system, the proper exiting of the runtime (e.g. over a function call) is delayed.

Exiting is also delayed while the SBO procedure is active. If SBO is active, it will be deactivated.



Information

This situation can arise especially for the action 'Pulse command' with watchdog or edge generation with one-step execution. Runtime is only ended once the action has been completed.