

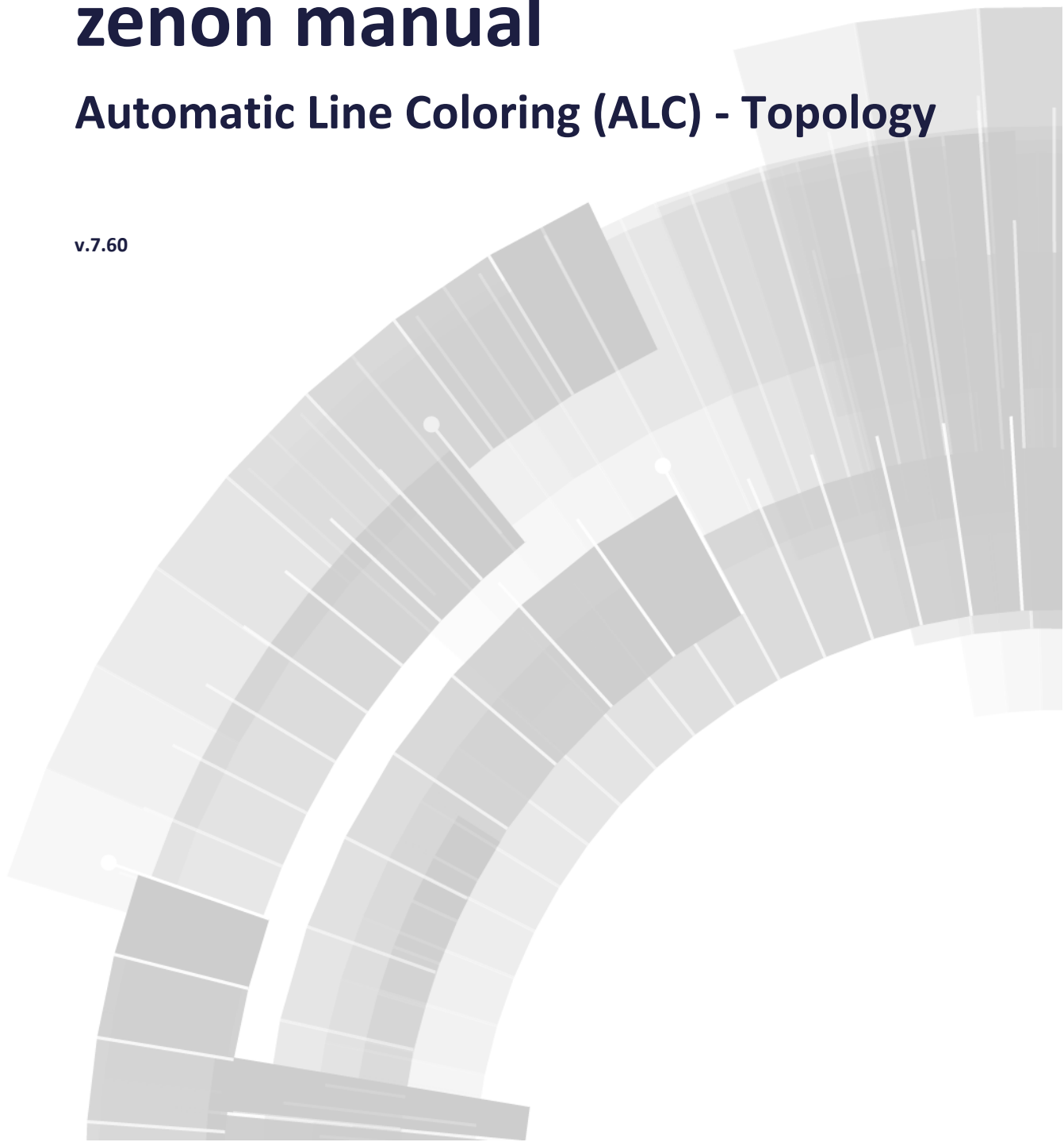


**COPADATA**  
do it your way

# zenon manual

## Automatic Line Coloring (ALC) - Topology

v.7.60





©2017 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.

# Contents

<b>1. Welcome to COPA-DATA help .....</b>	<b>5</b>
<b>2. Automatic Line Coloring (ALC) - Topology .....</b>	<b>5</b>
<b>3. ALC elements.....</b>	<b>7</b>
3.1 Procedural elements .....	8
3.1.1 Switch example - colors from ALC .....	13
3.1.2 Connection points of procedural elements .....	17
3.1.3 Switch input/output .....	19
3.2 Lines .....	19
3.2.1 Example .....	24
3.2.2 Connection points of lines .....	25
3.3 Checking the project .....	26
<b>4. Configuration .....</b>	<b>27</b>
4.1 Configuration of the sources.....	27
4.1.1 Coloring mode for UNDEFINED .....	31
4.2 Configuration of topological interlockings .....	32
4.2.1 Disconnecter under load - interlocking conditions .....	35
4.3 Configuration of the screen marker .....	36
<b>5. Function: Change ALC source color .....</b>	<b>38</b>
<b>6. Alias for detail screens .....</b>	<b>40</b>
<b>7. Fault locating in electric grids.....</b>	<b>43</b>
7.1 Search for ground fault .....	44
7.1.1 Mode of the search for ground faults .....	45
7.1.2 Ground fault detection type.....	46
7.1.3 Ground fault display .....	47
7.1.4 Earth fault triggering .....	47
7.1.5 Start search for ground fault .....	48
7.1.6 Acknowledge ground fault message .....	49
7.1.7 Stop search for ground fault.....	50

7.2	Short circuit search .....	51
7.2.1	Ground fault detection .....	52
7.2.2	Ground fault display .....	52
7.2.3	Ground fault detection triggering .....	53
7.2.4	Acknowledge short-circuit message .....	54
<b>8.</b>	<b>Impedance-based error detection and calculation of load distribution .....</b>	<b>55</b>
8.1	Impedance-based fault location of the short circuit .....	55
8.1.1	Engineering in the Editor .....	56
8.2	Expanded topological model .....	57
8.3	API .....	59

# 1. Welcome to COPA-DATA help

## ZENON VIDEO-TUTORIALS

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

## GENERAL HELP

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

# 2. 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 40) 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.
- ▶ Topology package: Requires additional licensing on the server (not on the client) and expands ALC by:
  - Multiple supply
  - Secured supply
  - Topological interlockings
  - Transformer and separator topological elements
  - Error detection (version 6.50 and above)

### 3. ALC elements

Automatic Line Coloring (ALC) makes it possible to color lines depending on 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 8) (source, switch/disconnector, drain, transformer or link) and on the other hand lines (on page 19).

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.

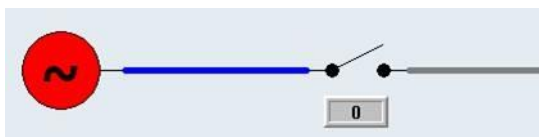
The different functions of the elements are assigned in the properties of the combined element.

#### EXAMPLE

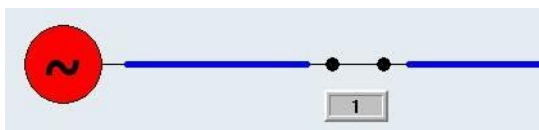
*A source has a connected line. A switch is connected to the line. And a second line is connected there. If the source is active, the first line is colored with the color of the Automatic Line Coloring defined in the source up to the valve. The other line is not colored before the switch is closed.*



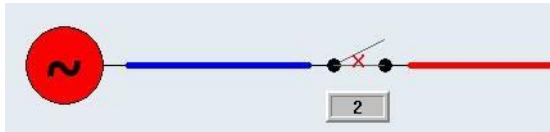
Source inactive



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 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
<b>Function type</b>	Defines the technological type of the Combined element.
Terminator	For bus bar ends. Blocks the error message " <b>Line only connected on one side</b> " 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. <b>Note:</b> 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><b>Attention:</b> 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 <b>Color from ALC</b> 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 27) 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 invalid (value: 3) or undefined (value: 2) or the status of the main variable is INVALID, the line colors itself in 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><b>Example:</b> see <b>Switch example - colors from ALC</b> (on page 13) section.</p> <p><b>Note:</b> If the <b>Switch input/output</b> property is active, the input and output of this element are reversed for the ALC.</p>
Valve	A slider (a valve) acts in a similar manner to a switch, but it is used for water and gas lines.

	<p>Value of the main variable:</p> <ul style="list-style-type: none"> <li>▶ Switch OFF: Value 0 -&gt; Slider closed-&gt; No forwarding</li> <li>▶ Slider ON: Value 1 -&gt; Slider open completely-&gt; Water flow</li> <li>▶ Slider DIF: Value 2 -&gt; Slider partially open-&gt; Water flow</li> <li>▶ Slider STO: Value 3 -&gt; Slider malfunction</li> </ul> <p><b>Note:</b> If the <b>Switch input/output</b> 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 are used to calculate the ALC - topological interlockings (in the <b>command processing</b>) <b>'Device would not be supplied'</b>.</p>
Check valve	<p>The <code>check valve</code> only forwards information in one direction.</p> <p>Value of the main variable:</p> <ul style="list-style-type: none"> <li>▶ Value 0: The forwarding is not active (= the valve is closed)</li> <li>▶ Value 1 or 2: Forwarding is only possible in one direction. In doing so, the color of the source is only forwarded from the input to the output. Forwarding in the opposite direction is not envisaged. This also affects the forwarding of the ground.</li> <li>▶ 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.</li> </ul> <p><b>Note:</b> If the <b>Switch input/output</b> 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><b>Note:</b> If the <b>Switch input/output</b> 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 <b>Source for reverse feed</b>, 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 <b>Source for reverse feed</b> property and not the <b>Source</b> property is used for further distribution of the source number.</p> <p><b>Note:</b> 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>. However, a disconnecter in the topological model must not be switched when live - topological interlocking <b>"Disconnecter under load"</b> in the command processing.</p> <p>As with the switch, the main variable determines the status: <code>On</code>, <code>off</code>, <code>intermediate position</code>, <code>malfunction</code>.</p> <p><b>Note:</b> If the <b>Switch input/output</b> property is active, the input and output of this element are reversed for the ALC.</p>

The source numbers given - for the source and transformer function types - are forwarded via closed switches (disconnectors, sliders etc.) up to the devices (drains). The colors of all connected lines and process-related elements are calculated from the higher-level sum of the supplying source numbers.

## SOURCE AND LINK NUMBER

Parameter	Description
<b>Source</b>	<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 27) chapter.</p> <p><b>Attention:</b> the pre-defined system sources (ID 0..9) are not suitable for your own sources in the project; exception - ground.</p>
<b>Link number</b>	<p>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 <code>Link</code>.</p> <p>This property is only active, if the function type <code>link</code> has been selected.</p>

## VARIABLES OF PROCESS-RELATED ELEMENTS

In order for a switch (and disconnecter, slider, etc.) to be given the status - open, closed, invalid - a BOOL data type or integer variable must be linked in the respective combined element as the main variable.

### Example:

- IEC870 driver: Variables with **Typ ID** `T01..T37`
- IEC850 driver: Variables `*/Pos/stVal[ST]`
- DNP3 driver: Input Variables

Pre-requisite: the **DPI/DPC mapping** has not been deactivated in the driver.



### Information

*For the position of a switch, only the first two bits of the main variable are taken into account.*

- ▶ The first bit is the actual switching; 0 is OFF and 1 is ON.
- ▶ The second bit is the error bit. There is no error if it is 0.

For a source, the status - "present" (ON) / "not present" (OFF) - must also be evaluated using the linked main variable. The internal BOOL data type variables are best suited to this. The source can then be

linked to the rest of the topology using a switch or a disconnecter (as is common in reality), so that only the position of the switch forwards the color of the source.

**Note:** for the main variable of a source that is connected to the network by means of a switch/disconnector, such as ground for example, create an **internal driver** variable, configure it with **calculation** `network` and **initial value** 1 ("always present").0 Alternatively, you can also link a source to a process variable directly (the source and its switch in one) if you do not need any **topological interlockings** when switching the source.

## STATES

- ▶ A switch and a source are switched on (closed) 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 bit. 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 main 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 using subsequent closed switches.



### Attention

*If, in the individual statuses of the combined element, the color and the fill color from ALC is activated, it is not just the line, but also the process-related elements that are colored in Runtime.*

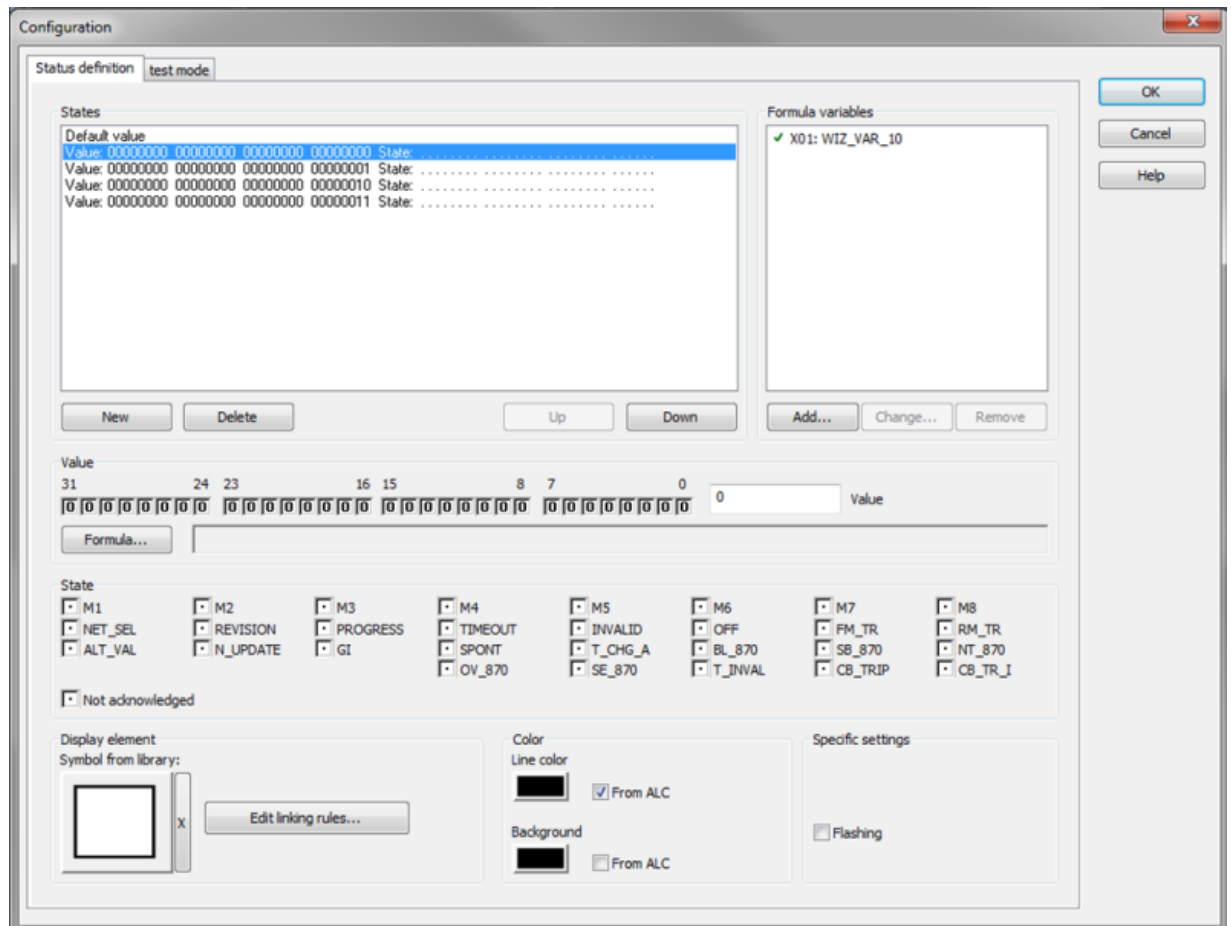
## 3.1.1 Switch example - colors from ALC

### 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)

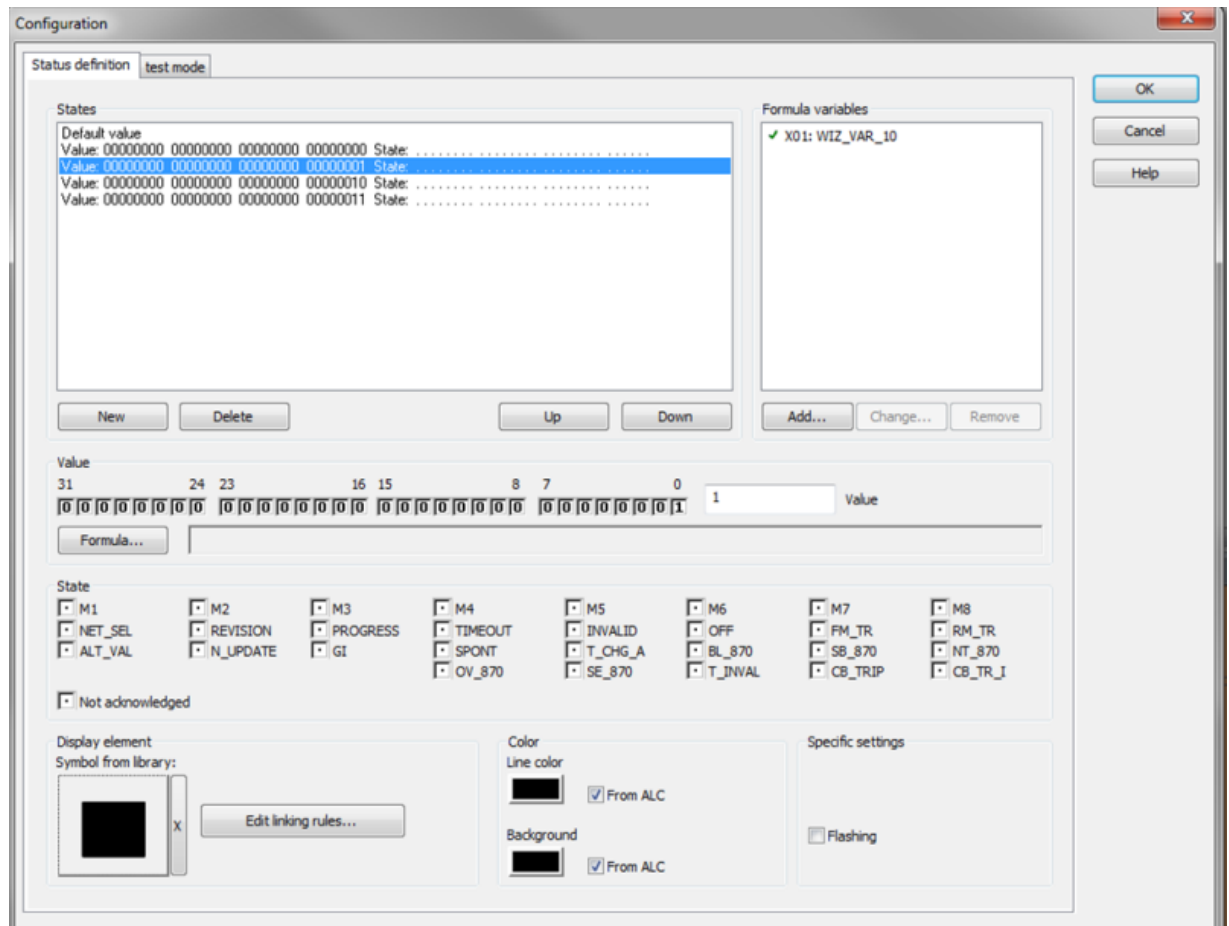


## EXAMPLE 2

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)



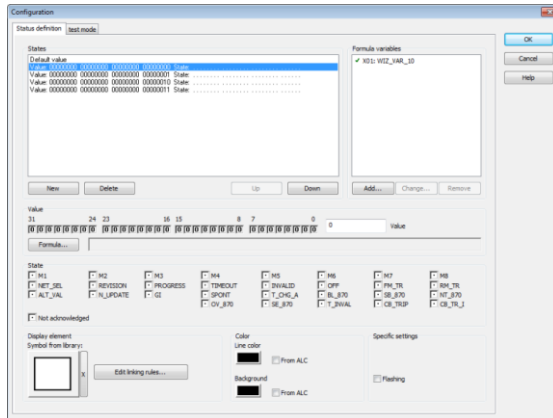
### EXAMPLE 3

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:

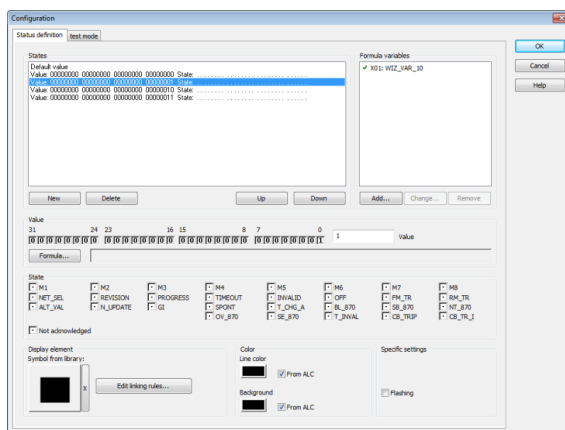
- Source color: Green
- Color not energized and construction color of the line: White
- Defined line and fill color of the combined element: black
- Switch status: off/open (value 0)



## 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 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: on/closed (value1)



### 3.1.2 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

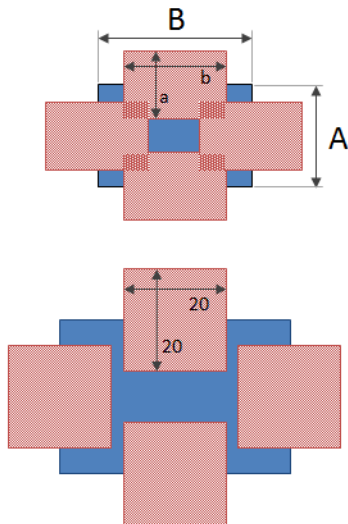
- ▶ The connection area for a connection point is in the middle of each side of the combined element. Each combined element thus has four connection points.
- ▶ The size of a connection area corresponds to 2/3 of the height and width of a combined element, but no more than 20 pixels.
- ▶ Each connection 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.

### 3.1.3 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

## 3.2 Lines

Lines are represented by the vector elements Line, Polylines and Pipe.

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:

Parameter	Description
<b>Color from ALC</b>	Activates the automatic line coloring for this vector element. 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.
<b>Priority for display</b>	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"> <li>▶ has <code>multiple supplies</code> and a <code>secured supply</code>,</li> <li>▶ The priority is set to <code>No priority</code>,</li> <li>▶ The display for <code>multiple supply</code> is set to <code>two sources with highest priority</code>,</li> <li>▶ The display for <code>secured supply</code> is set to <code>double width</code>,</li> </ul> Then the line is twice as wide and displayed as a dashed line in two colors.
<b>display multiple supplies</b>	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. <b>Note:</b> 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 <b>Dashing length supplied multiple times</b> 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 <b>Alternative color</b> property is used.
<b>Dashing length supplied multiple times</b>	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"> <li>▶ Minimum: 0 (automatic dash length)</li> <li>▶ Maximum: 32767</li> <li>▶ Default: 0</li> </ul>
<b>Alternative color</b>	Alternative color for the ALC coloring of lines, polylines or tubes with multiple supplies.
<b>display secured supply</b>	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 always 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"> <li>1. The defined RGB color is transformed to the HLS system.</li> <li>2. L (luminance = brightness) is recalculated with <math>\text{NewLuminance} = 240 \cdot 3/4 + L/4</math></li> <li>3. The color value is recalculated to the RGB system with the new brightness.</li> </ol> <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.
<b>Use alias</b>	Active: Alias is used.
<b>Alias</b>	Opens the dialog (on page 40) 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:


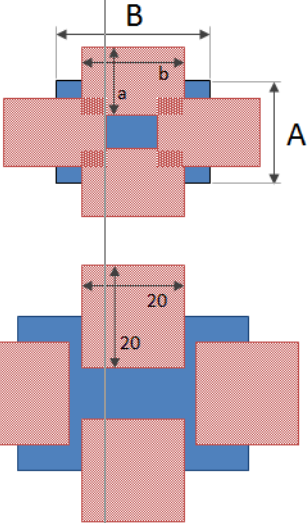






1. Automatic Line Coloring (highest priority, overrules all other settings)
2. Dynamic colors
3. Static colors

3.2.1 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		



### 3.2.2 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.

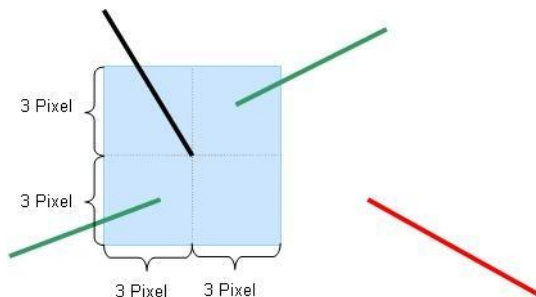


#### Example

*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

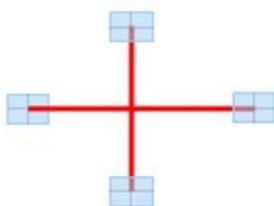
*Any number of lines can be connected in a connection area.*



#### 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.*

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

## 4. 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 27)

Create new sources - in the desired colors for the topology.

**Note:** In doing so, note that the system sources (ID 0..9) already have a pre-defined meaning or are reserved for future versions.

Note also the principles for coloring for UNDEFINED (on page 31).

- Interlockings (on page 32)

Configure which **topological interlockings** the **Command Processing** module should take into account.

**Note:** the tab is only visible with the topology package license.

- Screen marker (on page 36)

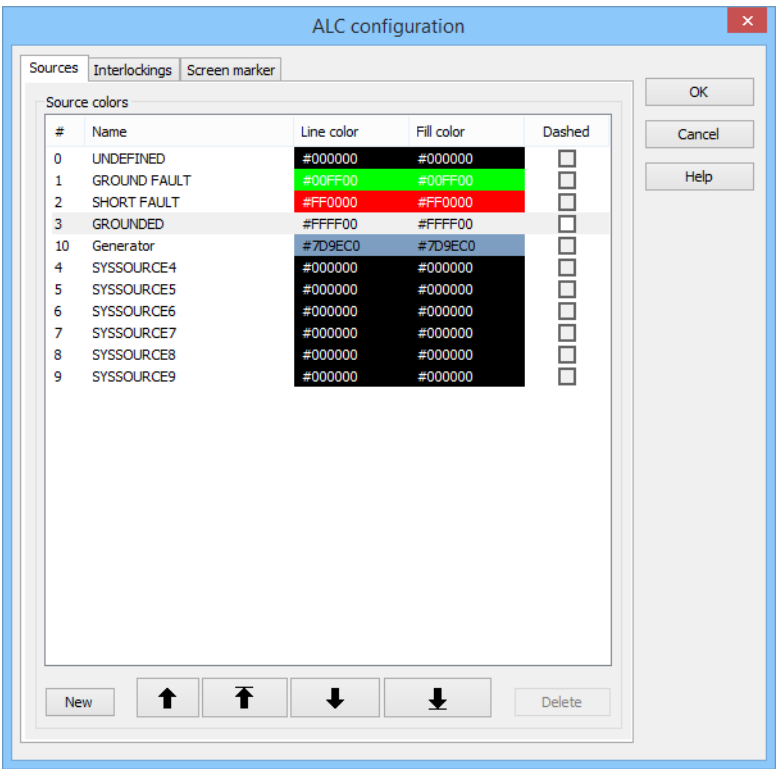
Configure the color table for the screen marker for **impedance-based error detection**.

**Note:** the tab is only visible with the topology package license.

### 4.1 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'. Sources with ID between 0 and 9 are reserved for system sources. Those that already have a function (such as GROUNDED - the color of the "earth" source) must not have their function changed. Those that do not yet have any functionality in the current zenon version remain reserved for future versions.

The source colors from ID #10 are freely available for the process-related elements, for example source: "Generator" or "110kV" etc. Add further colors for this.



## SOURCE COLORS

Parameter	Description
<b>Number</b>	Internal unique consecutive number, so that the source can be identified. This number is given by the system automatically and cannot be changed.  <b>Attention:</b> IDs 0 to 9 are reserved for the system sources and must not be used user-specific.
<b>Name</b>	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.  <b>Note:</b> The labels are not language switchable.
<b>Line color</b>	Line color of the respective source. This color is used for coloring lines, polylines and as the outside color of tubes.
<b>Dashed</b>	Check box for activation. If active, the line is drawn as dashed.  <b>Note:</b> This checkbox can only be activated for <b>GROUNDED</b> . This check box is grayed out for all other sources.
<b>New</b>	Adds a new color.
<b>Delete</b>	Deletes the selected color.
<b>Upwards (arrow symbol)</b>	Moves selected source up one position.
<b>Fully upwards (arrow symbol)</b>	Moves selected source to the start of the list.
<b>Downwards (arrow symbol)</b>	Moves selected variable down one position.
<b>Fully downwards (arrow symbol)</b>	Moves selected source to the end of the list.

## CLOSE DIALOG

Options	Description
<b>OK</b>	Applies all changes in all tabs and closes the dialog.
<b>Cancel</b>	Discards all changes in all tabs and closes the dialog.
<b>Help</b>	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.  
The field is switched to editing mode.
2. Enter the code.
3. Press the `Enter` key or select another source to apply the change.

To select via a color palette:

1. highlight the desired line.
2. Click on the `...` button behind the color  
**Note:** The `...` button is only visible if the color entry is selected with a mouse click.  
 The color palette is opened in the context menu.
3. select the desired color

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

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

## 4.1.1 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.

## 4.2 Configuration of topological interlockings

In Runtime, the **Command Processing** modules can calculate the interlockings that result from the dynamic status of the electrical network, the topology of which was configured with **ALC**, from scratch. Using the topology of the network configured with ALC and current statuses of the sources (`ON / OFF`), switches, disconnectors etc. can automatically detect the **command input** that will result in the execution of a command, for example "Voltage to ground" and then prevent the execution of the command.

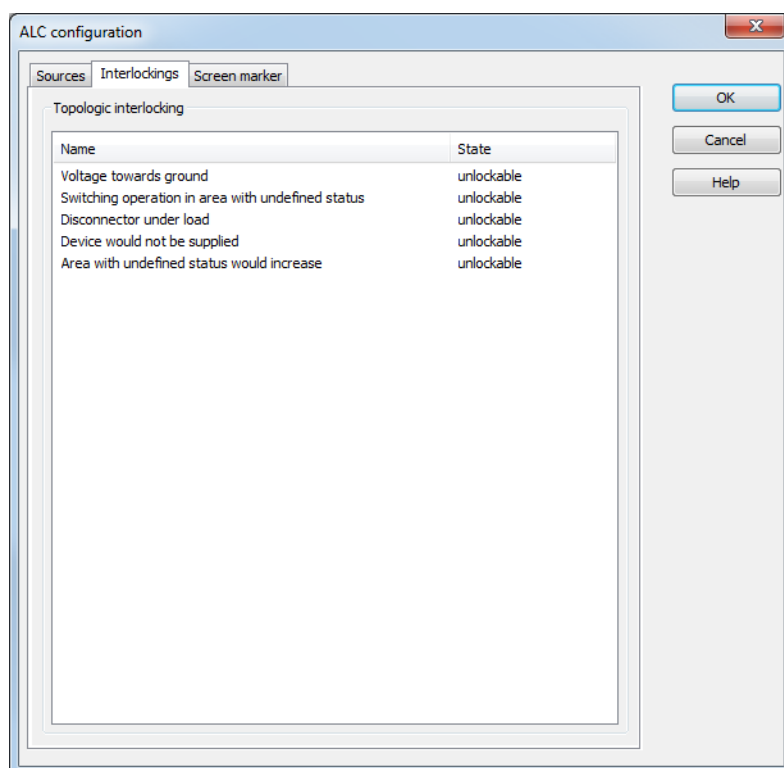
The topological interlockings from the **ALC** for **command input** are configured centrally - for the respective project. In doing so, a decision is also made as to whether a user can unlock an interlocking (provided they also have the **authorization level for unlocking** for the action).





## Information

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



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

Parameter	Description
<b>Voltage towards ground</b>	Interlocking is active if a <code>switch/disconnector</code> is to be closed and a grounded potential is connected to its first connector and its other connector is connected or undefined.
<b>Switching action in an area with an undefined status</b>	Interlocking is active if a <code>switch/disconnector</code> is to be closed and both of its connectors are 'undefined' or 'disturbed'.
<b>Disconnecter under load</b>	Interlocking is active if certain conditions have been met for switching the <code>disconnecter</code> on or off.  Conditions: See "Disconnecter under load - interlocking conditions (on page 35)" section.
<b>Device would not be supplied</b>	Interlocking is active if a <code>switch/disconnector</code> is to be opened and a device that is switched on and supplied with voltage from a <code>source (drain)</code> then loses supply.
<b>Area with undefined status would increase</b>	Interlocking is active if a <code>switch/disconnector</code> is to be closed and one connector is 'undefined' or 'disturbed' and the other not.  It is also reported if the <code>command</code> has been configured with the <b>switching direction</b> <code>none</code> .

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

Parameter	Description
<b>do not check</b>	The selected condition is not considered in this project (topological model).
<b>unlockable</b>	The selected condition is considered in this project. If the condition applies, the user can unlock it with the Command Processing (in the screen of type <b>Command Processing</b> ). This unlocking action is logged in the <b>Chronological Event List</b> .
<b>not unlockable</b>	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.

### 4.2.1 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 for the line segments that connect the disconnectors:

#### WHEN TURNING THE DISCONNECTOR ON:

A check is carried out to see whether the topology before switching to **ON** is in one of the following states:

- ▶ Both line segments are supplied/grounded by the same source;
- ▶ One line segment does not receive any voltage and the other line segment is grounded;
- ▶ A line segment is not under load.

#### WHEN TURNING THE DISCONNECTOR OFF:

A check is carried out to see whether the topology after switching to **OFF** is in one of the following states:

- ▶ Both line segments are supplied by the same source;
- ▶ One line segment stops receiving voltage, the other line segment is grounded;
- ▶ A line segment stops being under load.



#### 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:  
Switches and disconnectors connected to the line segment are closed but only connect to one additional segment that is also not under load.

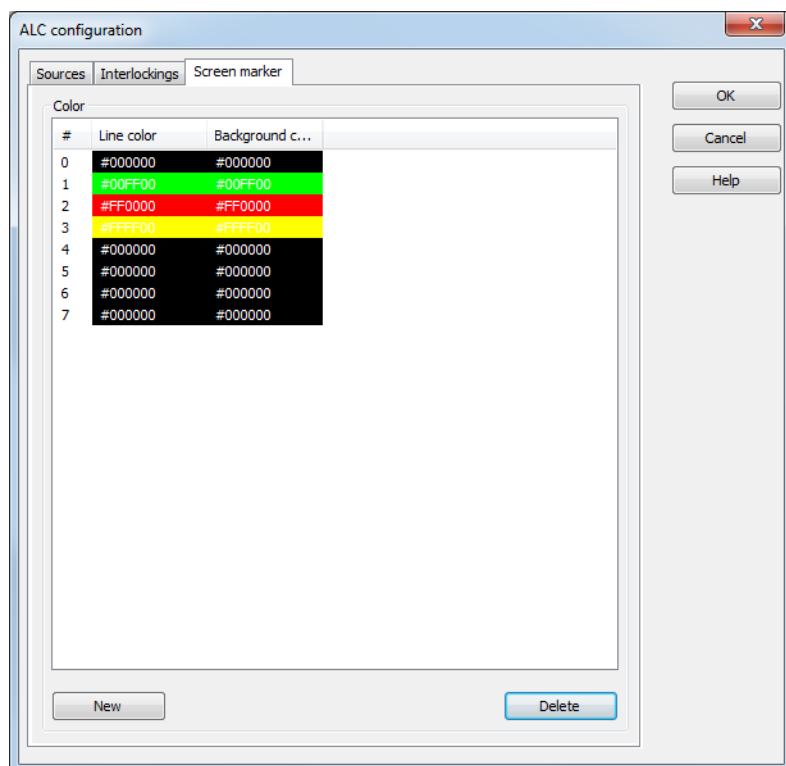
In addition, all of the following conditions must be met for the status of `not under load`:

- ▶ 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 (one open line).

▶

## 4.3 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 59). See also: **AddMarker**.



Parameter	Description
<b>Number</b>	Unique internal serial number for clear assignment. This number is given by the system automatically and cannot be changed.
<b>Line color</b>	Line color of the screen marker.
<b>Fill color</b>	Fill color of the screen marker.
<b>New</b>	Adds a new color.
<b>Delete</b>	Deletes the selected color. <b>Note:</b> Only the last color in the list can be deleted. Standard colors cannot be deleted.

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.  
The field is switched to editing mode.
2. Enter the code.
3. Press the `Enter` key or select another source to apply the change.

To select via a color palette:

1. highlight the desired line.
2. Click on the `...` button behind the color  
**Note:** The `...` button is only visible if the color entry is selected with a mouse click.  
The color palette is opened in the context menu.
3. select the desired color

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

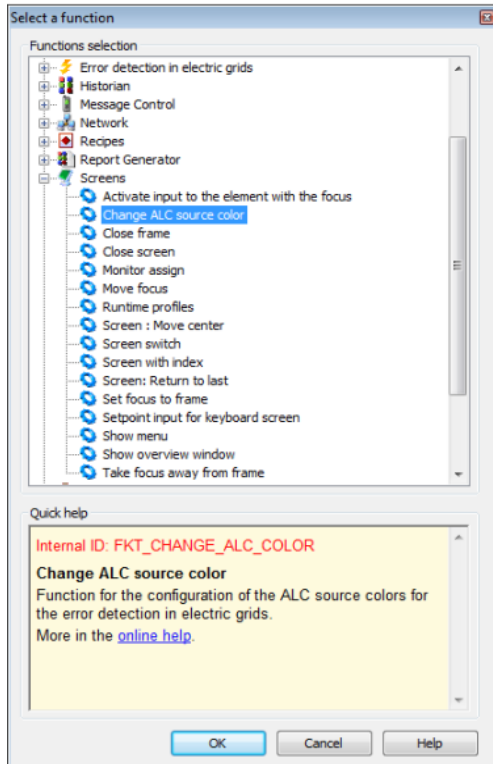
Item	Meaning
#	Identifier to indicate that a hexadecimal color code is used.
RR	2 digits are the red value of the color in hexadecimal system. 0-255 corresponds to 0-FF.
GG	2 digits are the green value of the color in hexadecimal system. 0-255 corresponds to 0-FF.
BB	2 digits are the blue value of the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

## 5. Function: Change ALC source color

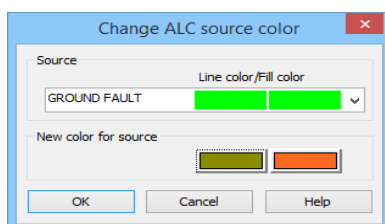
The foreground and background color of an ALC source can be temporarily changed for the coloring in Runtime using the **Change ALC source color** function. The change remains until Runtime is ended, reloaded or the function is executed again. To create the function:

- ▶ 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.

## 6. Alias for 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.

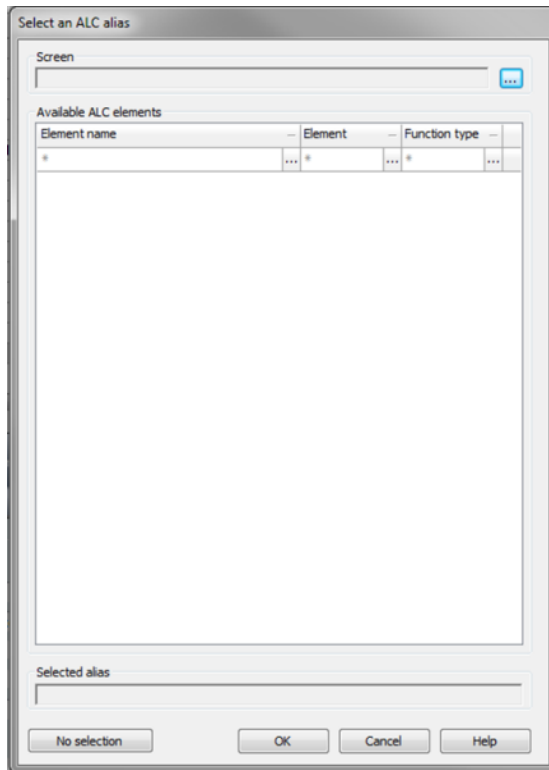
Solution: Replace the period in the screen name with a different character, such as an underscore for example (\_).

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.



Parameter	Description
<b>Screen</b>	Click the ... button and a dialog opens to select a screen.
<b>Available ALC elements</b>	<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><b>Filter</b></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"> <li>▶ <b>Name:</b> Input of a standard search term with wild cards (*). The last 12 search terms are offered in the list until the Editor is ended.</li> <li>▶ <b>Element:</b> Select from drop-down list.</li> <li>▶ <b>Function type:</b> Select from drop-down list.</li> </ul> <p>Clicking on ... opens saved search or drop-down list.</p> <p>If a filter is active, clicking on the <b>X</b> deletes the filter.</p>
<b>Selected alias</b>	Shows the selected element in the field of <b>Available ALC elements</b> .
<b>No selection</b>	Removes selected element.
<b>OK</b>	Saves selection and closes dialog.
<b>Cancel</b>	Discards changes and closes dialog.
<b>Help</b>	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.

## 7. 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. Starting points for error detection are called ground fault or short circuit reporters (such as a protective device) that are assigned to a circuit breaker. It is assumed that the ground fault and short circuit reporters are always at the output of the circuit breaker element. For this reason, the corresponding variables (with reports from the protective device) are linked to ALC `switch` elements.

The reports from protective devices are shown by means of special coloring - with the source colors ID 1 and 2, but only if the report is received whilst the lines are live. At the same time as this, the reports are set to the additional variables for display. Graphic error displays can thus also be displayed in the screen, for example with further combined elements that are only visible if there is a display active.

The display must be reset manually (acknowledged) once the protective devices have retracted the reports.



### 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 on each computer in the zenon network. 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 topology.

Error detection in the electrical network is divided into:

- ▶ Search for ground fault (on page 44)
- ▶ Short circuit search (on page 51)

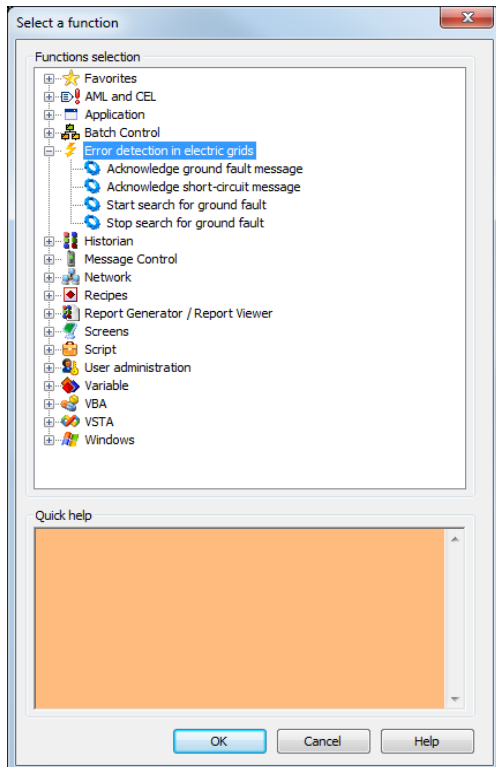
To configure error detection

- ▶ You require a license for ALC and zenon Energy Edition
- ▶ configure the appropriate screens
- ▶ Configure (on page 8) ALC to the corresponding combined elements with the `switch` function
- ▶ configure (on page 19) 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 48)
- ▶ acknowledge (on page 49) ground fault message (on page 49)
- ▶ Stop search for ground fault (on page 50)

- Acknowledge short-circuit message (on page 54)



## COLORINGS

Errors can be displayed with special coloring of the lines in the ALC if the notifications are received whilst the lines are live. 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 38) 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

## 7.1 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** (ID 1) source. At the same time as this, the notifications are set to the additional variables for graphical **display**.

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 the notification of the direction comes in **forwards** and/or **backwards**, then via the **trigger**.
  - Lagging: First comes the **trigger**, then the solution.



#### 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 46)
2. Define the mode of search for ground fault (on page 45), ground fault trigger (on page 47) and ground fault display (on page 47).
3. Create the functions for start search for ground fault (on page 48), acknowledge ground fault report (on page 49) and end search for ground fault (on page 50)



#### 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.*

### 7.1.1 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.

### 7.1.2 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 48) 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.*

## 7.1.3 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.

## 7.1.4 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.

### 7.1.5 Start search for ground fault

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

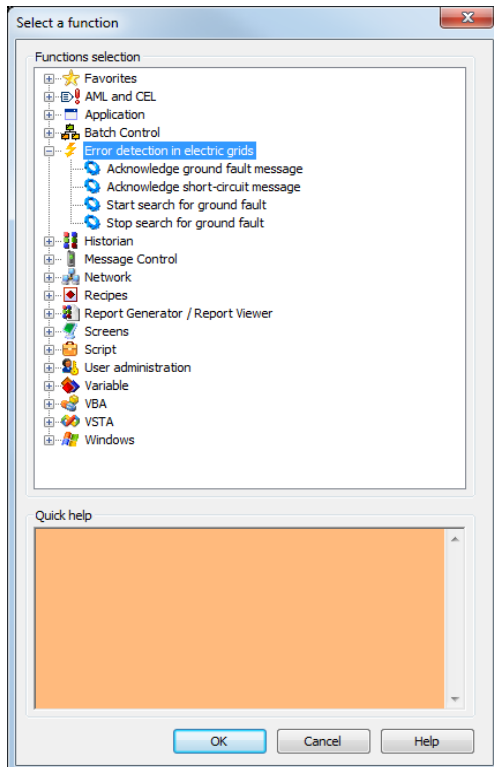
1. Fault reports from all ground fault identification devices that were configured with wiper message processing are ignored.
2. The search algorithm is changed: Switch actions can only reduce the area subject to a ground fault further. Newly received messages do not therefore increase the area potentially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

### 7.1.6 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.

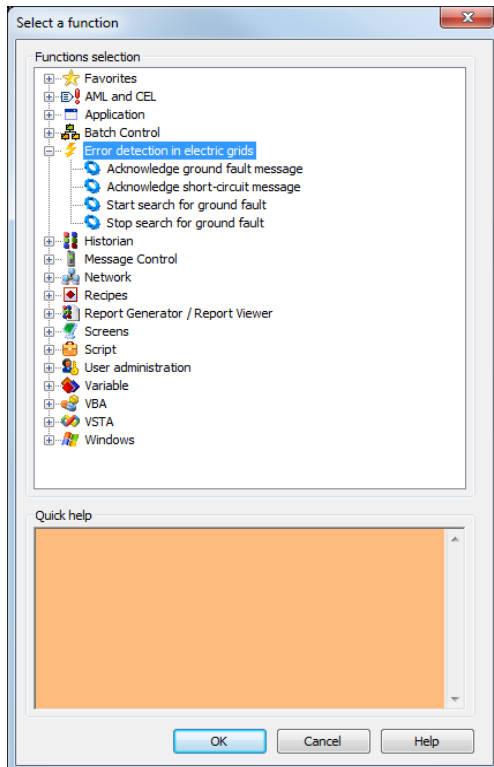
Rules when acknowledging:

- 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

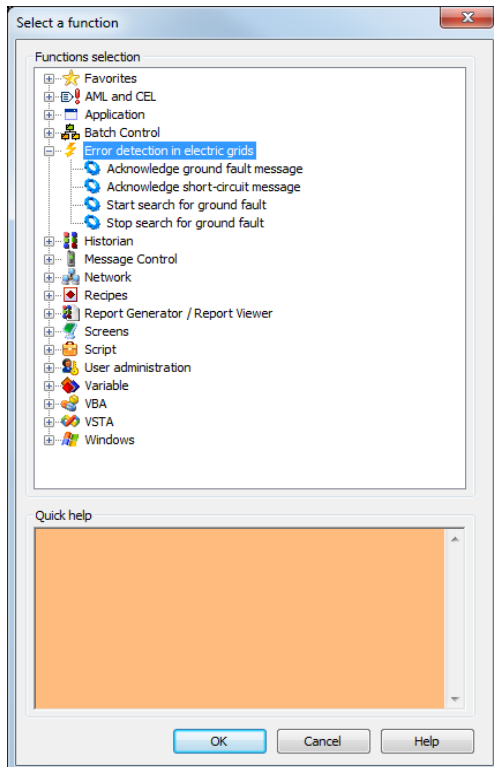
### 7.1.7 Stop search for ground fault

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

## 7.2 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 52)

2. Define ground fault display (on page 52) and triggering of ground fault detection (on page 53)
3. Set up the function for acknowledgment of ground fault message (on page 54)

### 7.2.1 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

### 7.2.2 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

### 7.2.3 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

### 7.2.4 Acknowledge short-circuit message

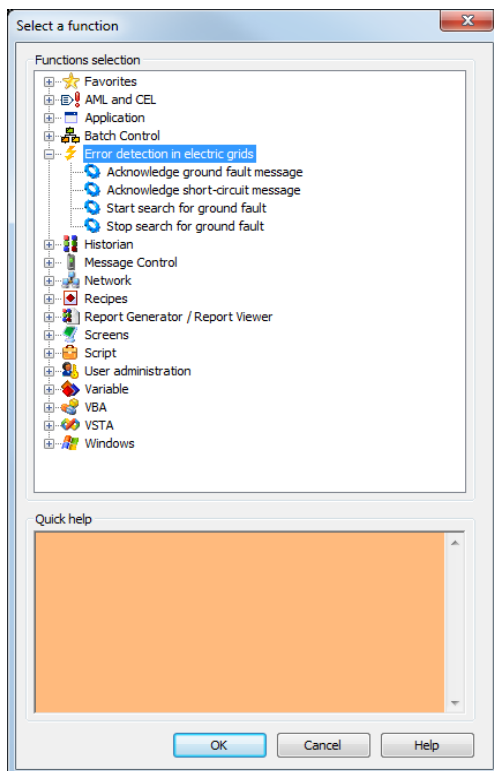
With the **Acknowledge short-circuit message** function, an internally recorded short circuit from a short circuit indication device can be acknowledged. In doing so, the internally-latched ground fault status is reset if the status is still pending, or highlighted as acknowledged. A recorded short circuit 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

## 8. 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. Fault locating from protection is possible by means of configuration in the zenon Editor.

The model provides properties and methods for external evaluation of the fault location and load distribution via API.

### PROPERTIES FOR ALC AND THE EXTENDED TOPOLOGICAL MODEL

The ALC elements **combined element** and **line** (line, polyline, tube) have special properties for impedance-based fault location and to calculate the load distribution. The properties for load distribution are not evaluated in zenon, but are available via the zenon API as algorithms to be created by users.

The simple topological model for the coloring is 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

### 8.1 Impedance-based fault location of the short circuit

With impedance-based fault location, an error marker is set at the location of the problem in the topology. The impedance values measured by protective devices are evaluated by the **ALC** module. Based on the topology, the error markers are positioned in the screen correctly in a zenon screen.

If a short circuit occurs and the reactance is not equal to zero, the search for the location of the short circuit starts:

- ▶ Short circuit:  
Reported by a linked variable for the **Triggering** property
- ▶ Reactance:  
Value of the variable (from the REAL data type), that is linked to the **Reactance value from protection** property.

## POSITION OF THE MARKER

All lines are run through in the corresponding direction. The direction results from negative or positive reactancy. The respective reactancy of the line run through is deducted and the search continues until the residual reactancy is less than the reactancy of the next line. A marker is drawn in the line. The position of the marker corresponds to the residual reactancy.

If there is no reactancy value, no marker is set in the event of a short circuit notification. In order for the marker to be drawn correctly, the area must not be under load during the short circuit notification. With lagging short-circuit notifications, the reactancy is only evaluated if the notification of direction has been received or the timeout of 2 seconds has expired.

The search is canceled if an open shift element or another ALC element has been found. Each part of the network and each individual line therein must only run once per trigger, there are thus less markers that occur in the line network than would be possible.

When reloading, markers that already exist are drawn at the same point as before reloading. Changes to the configuration of the fault locating are only evaluated after another short circuit.

If a short circuit notification is removed and acknowledged, all markers of this short circuit trigger are deleted.

**Note:** Depending on the order of the rectification of the short circuit and switching on again, marker can remain drawn in, although the line is no longer colored as a short circuit.

### 8.1.1 Engineering in the Editor

With impedance-based fault location, an error marker is set at the location of the problem in the topology. The location is calculated from impedance, on the basis of the topology.

To configure the impedance-based fault location in the zenon Editor, carry out the following steps:

1. Activate impedance-based fault location:
  - a) To do this, click on the project in your **Workspace**.
  - b) Click on the **Automatic Line Coloring** project property group.
  - c) Activate the **Fault location based on impedance** property.
2. Configure the display of the screen markers with the project properties:
  - a) **Screen marker size**
  - b) **Line width of the screen marker**
  - c) **Display type of the screen marker**
3. Create a zenon screen.
4. Position the **combined element** on the zenon screen.  
The variable selection dialog is opened.



5. Configure the ALC settings for the combined element:
  - a) Ensure that the combined element has been selected.
  - b) Switch to the **Automatic Line Coloring** property group.
  - c) In the **Function type** property, select the `Switch` entry from the drop-down list.
  - d) Link the **Reactance value from protection** property (in the **Fault location from protection/load distribution** properties section) to a `REAL` data type variable with the value of the measured impedance.
  - e) Select the type of **Short-circuit detection** in the drop-down list of the **Type** property.
  - f) Configure the color of the marker in the **Marker color** property.

## 8.2 Expanded topological model

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"

## 8.3 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 screens via the zenon API. These marker elements are available for the following elements:

- ▶ Line
- ▶ Polyline
- ▶ Pipe

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