zenon manual
Interlockings

v.8.00
# Contents

1. Welcome to COPA-DATA help .......................................................................................... 4

2. Interlockings .................................................................................................................. 4

3. Detail view of context menu and toolbar ....................................................................... 5

4. Engineering in the Editor ............................................................................................... 8
   4.1 Creating Interlockings ............................................................................................... 8
      4.1.1 Substitution of interlocking variables .................................................................. 10
      4.1.2 Example of configuration to substitute interlocking variables ......................... 14
   4.2 Formula editor .......................................................................................................... 16
      4.2.1 List of status bits .............................................................................................. 20
      4.2.2 Logical operators ............................................................................................ 22
      4.2.3 Bit formulas ..................................................................................................... 23
      4.2.4 Comparison operators ..................................................................................... 24
      4.2.5 Examples for formulas ..................................................................................... 25

5. Operating during Runtime ............................................................................................ 26
   5.1 Usage of Interlockings .............................................................................................. 26
1. Welcome to COPA-DATA help

ZENON VIDEO-TUTORIALS

You can find practical examples for project configuration with zenon in our YouTube channel. The tutorials are grouped according to topics and give an initial insight into working with different zenon modules. All tutorials are available in English.

GENERAL HELP

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

PROJECT SUPPORT

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

LICENSES AND MODULES

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

2. Interlockings

Interlockings control the access to certain zenon objects in Runtime using variables. Operation can be blocked or released depending on variables. Depending on process statuses, operations can be activated/deactivated.

It is possible to create and use various interlockings within one project. In each interlocking several parallel interlocking conditions are possible.
INTERLOCKING OF OBJECTS

All dynamic elements except trend element and message element.

PROJECT MANAGER CONTEXT MENU

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New interlocking</td>
<td>Creates a new interlocking and opens the dialog for selecting variables.</td>
</tr>
<tr>
<td>Export all as XML</td>
<td>Exports all entries of the interlocking as an XML file.</td>
</tr>
<tr>
<td>Import XML</td>
<td>Imports XML files.</td>
</tr>
<tr>
<td>Editor profile</td>
<td>Opens the drop-down list with predefined editor profiles.</td>
</tr>
<tr>
<td>Help</td>
<td>Opens online help.</td>
</tr>
</tbody>
</table>

Information

Variables for the interlocking can be replaced. For more details see chapter Substitution of variables and functions in dynamic elements.

Example

A machine is in full automatic operation, so it should not be switched to manual operation. Only if a certain operation status (e.g. STOP) is reached, it is allowed to be switched. With an interlocking the button for switching between manual and automatic operation can be locked in the visualization surface, until this status (e.g. STOP) is true. This can avoid incorrect operation.

3. Detail view of context menu and toolbar

TOOLBAR
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New interlocking</strong></td>
<td>Creates a new interlocking and opens the dialog for selecting variables.</td>
</tr>
<tr>
<td><strong>New interlocking condition</strong></td>
<td>Creates a new interlocking condition.</td>
</tr>
<tr>
<td><strong>Add variable</strong></td>
<td>Opens the dialog for selecting variables.</td>
</tr>
<tr>
<td><strong>Copy</strong></td>
<td>Copies the selected condition.</td>
</tr>
<tr>
<td><strong>Paste</strong></td>
<td>Pastes the condition from the clipboard.</td>
</tr>
<tr>
<td><strong>Delete</strong></td>
<td>Deletes selected condition.</td>
</tr>
<tr>
<td><strong>Export selected as XML...</strong></td>
<td>Exports selected entries as an XML file.</td>
</tr>
<tr>
<td><strong>Import XML</strong></td>
<td>Imports from an XML file.</td>
</tr>
<tr>
<td><strong>Rename</strong></td>
<td>Enables the element to be renamed. Also possible by clicking in the field</td>
</tr>
<tr>
<td></td>
<td>with the mouse or by pressing the F2 key.</td>
</tr>
<tr>
<td><strong>Properties</strong></td>
<td>Opens the property window for the selected element.</td>
</tr>
<tr>
<td><strong>Help</strong></td>
<td>Opens online help.</td>
</tr>
</tbody>
</table>

### CONTEXT MENU INTERLOCKINGS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New interlocking</strong></td>
<td>Creates a new interlocking and opens the dialog for selecting variables.</td>
</tr>
<tr>
<td><strong>Paste</strong></td>
<td>Pastes the interlocking from the clipboard.</td>
</tr>
<tr>
<td><strong>Export all as XML</strong></td>
<td>Exports all entries of the interlocking as an XML file.</td>
</tr>
<tr>
<td><strong>Import XML</strong></td>
<td>Imports XML files.</td>
</tr>
<tr>
<td><strong>Help</strong></td>
<td>Opens online help.</td>
</tr>
</tbody>
</table>

### CONTEXT MENU INDIVIDUAL INTERLOCKING

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add variable</strong></td>
<td>Opens the dialog for selecting a variable.</td>
</tr>
<tr>
<td><strong>New interlocking condition</strong></td>
<td>Creates a new interlocking condition.</td>
</tr>
<tr>
<td><strong>Copy</strong></td>
<td>Copies the selected interlocking.</td>
</tr>
</tbody>
</table>
**Paste**  
Pastes the interlocking from the clipboard.

**Delete**  
deletes selected interlocking.

**Export all as XML**  
Exports all entries as an XML file.

**Import XML**  
Imports from an XML file.

**Rename**  
Enables the element to be renamed. Also possible by clicking in the field with the mouse or by pressing the **F2** key.

**Properties**  
Opens the property window for the selected element.

**Help**  
Opens online help.

---

**CONTEXT MENU GROUP VARIABLES**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Add variable</strong></td>
<td>Opens the dialog for selecting variables.</td>
</tr>
<tr>
<td><strong>Paste</strong></td>
<td>Pastes the condition from the clipboard.</td>
</tr>
<tr>
<td><strong>Help</strong></td>
<td>Opens online help.</td>
</tr>
</tbody>
</table>

---

**CONTEXT MENU INDIVIDUAL VARIABLE**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Remove variable</strong></td>
<td>Deletes the selected variable after requesting confirmation.</td>
</tr>
<tr>
<td><strong>Copy</strong></td>
<td>Copies selected variable</td>
</tr>
<tr>
<td><strong>Paste</strong></td>
<td>Pastes the variables from the clipboard.</td>
</tr>
<tr>
<td><strong>Properties</strong></td>
<td>Opens the property window for the selected element.</td>
</tr>
<tr>
<td><strong>Help</strong></td>
<td>Opens online help.</td>
</tr>
</tbody>
</table>

---

**CONTEXT MENU INTERLOCKING CONDITIONS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New interlocking condition</strong></td>
<td>Creates a new interlocking condition.</td>
</tr>
<tr>
<td><strong>Paste</strong></td>
<td>Pastes the condition from the clipboard.</td>
</tr>
<tr>
<td><strong>Help</strong></td>
<td>Opens online help.</td>
</tr>
</tbody>
</table>
4. Engineering in the Editor

Configure the interlockings in the Editor in order to control access to certain zenon objects in Runtime using variables.

To do this, you can:

- Create new interlockings (on page 8)
- Substitute the interlocking variables of existing interlockings (on page 10) in order to be able to use an interlocking for as many applications as you want

4.1 Creating Interlockings

Create an interlocking with an interlocking variable. Then link these to a condition in order to be able to use the interlocking in Runtime.

**Note:** Variables of interlockings can be substituted for linked symbols and screen switching. Substitution of interlocking variables is only possible if, in the symbol properties under Linking rule, the checkbox of the Consider interlocking variables property is activated.

<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>A variable must not be an interlocking variable and a response variable at the same time. This configuration would lead to an infinite loop.</td>
</tr>
</tbody>
</table>

The interlocking can be linked to a condition. To do this, you must first create one or more new interlocking conditions in the properties window.

For each condition, a binary formula can be entered in the Logical link property.

**Note:** If a new interlocking is created, the following is entered into the interlocking conditions by default:

- Logical link property: <No formula>
- Interlocking text property: @<no interlocking text>

If a Interlocking text or a Logical link of a pre-existing interlocking is deleted, a warning message appears in the output window when compiling the project, stating that a Interlocking text or a Logical link is missing.

If only the Interlocking text of a pre-existing interlocking is deleted and the warning message is not heeded when compiling, no text is shown in the entry in the interlocking list if an action with precisely these settings is executed. The interlocking is active however.
The missing text in the entry of the interlocking list is attributable to the incorrect configuration of the interlocking.

Elements can be blocked in the visualization interface with interlocking.

**Information**

An element in the visualization surface is locked, if the interlocking condition applies, i.e. is logical 1 – TRUE.

**Interlocking:** If several conditions are defined in one interlocking, it is sufficient for locking the element if only one condition is fulfilled.

The formula editor (on page 16) can be used to define binary and numeric formulas as a locking condition. It is opened by clicking on the **Logical link** property in the properties window. The formula editor allows the definition of Binary statements with the help of the linked variables and **logical** or **bitwise** and **comparison operators**. See also chapter **Screens/Comparison operators** (on page 24).

**Information**

**Unlocking:** If several conditions are linked to the interlocking, all conditions that are met must be unlocked.

**CREATING AN INTERLOCKING**

1. Go to **Interlocking** in the project tree.
2. Go to **New Interlocking** in the toolbar or in the context menu.  
   The **variable selection** dialog is opened.
3. Select the desired interlocking variable with a mouse click.
4. Click on the **Add** button. 
   The interlocking variable is added.
5. Close the dialog by clicking on **OK**.  
   The selected interlocking variable is shown under variables. 
   A new condition is added.
6. You can change the name of the interlocking if you want. 
   a) To do this, left-click on the newly-created interlocking. 
      The properties window of the interlocking is opened. 
   b) Change the name under **General** and **Name**.
7. Right-click on **Interlocking Conditions** and **New Interlocking Condition**.
8. Click in the project properties under **Condition** and **Logical Linking** on the ... selection button. 
   The **formula** dialog is opened.
9. Please enter an interlocking condition.

   Example: \((X01.Value > 0)\)

10. Confirm the input by clicking on **OK**.

The creation of the interlocking is now complete.

**VALUES FOR RESPONSE VARIABLE**

The response variable provides values for the status of the interlocking:

- 0: not interlocked
- 1: interlocked
- 2: at least one condition variable does not have a value
- 3: at least one value for a condition has an **INVALID** bit

**4.1.1 Substitution of interlocking variables**

You can substitute interlocking variables in order to use a previously-created interlocking for many other applications, each with their own interlocking variable.

Example: **Interlocking 1** contains the **Interlocking 1** interlocking variable. The substituted interlocking variables **Interlocking 2, Interlocking 3, ...** then also use the settings of **Interlocking 1**.

**REQUIREMENTS**

- The symbol must have content with an interlocking (on page 8).
- In the symbol properties under **Linking rule**, the checkbox for the **Consider interlocking variables** property must be activated.
- The desired interlocking variables must have already been created under the Variables node.

**SUBSTITUTION OF INTERLOCKING VARIABLES WHEN LINKING SCREENS**

Engineering:

1. In the project tree under Screens, select the screen to which you want to link the symbol.
2. Open the desired screen with a double click.
3. In the project tree, go to **Screens** and **Symbol Library**.
4. Select the desired symbol with a mouse click.
Note: At least one piece of content of the symbol must have an interlocking in order to be able to substitute the interlocking variable.

5. Drag & drop the symbol to the desired screen to link the symbol to the screen.

The **Enter element** dialog is opened.

The interlocking variable is visible in the lower part of the dialog under **Preview Source** and **Target**.

Initially, **Source** and **Target** are filled with the same interlocking variable.

6. Enter, under **Linking Rule** and **Source (with '*' or '?')**, the name of the interlocking variable in Preview under Source.

**Example:** **Interlocking 1**

Under **Replace variables using**, **Name** and **Consider interlocking variables** are selected by default.

7. Enter the name of the new interlocking variable under **Interlocking rule** and **Replace with**.

**Example:** **Interlocking 2** The variable must have already been created beforehand.

8. In the **element entry** dialog, click on the **Process** button.

The number of replaced connections is shown in a separate dialog.

9. Close the dialog by clicking on **OK**.
The substituted interlocking variable of the content of the symbol is now visible under Preview and Target.

10. Close the dialog by clicking on OK.

**SUBSTITUTION OF INTERLOCKING VARIABLES IN THE SYMBOL LIBRARY**

Engineering:

1. In the project tree, go to the **Screens** and **Symbol Library** nodes.

2. Double-click on the symbol whose interlocking variables are to be substituted in order to selected.

   A view of the symbol is opened.

3. In the symbol, click on the linked symbol with the interlocking whose interlocking variables are to be substituted.

4. In the properties of the content of the symbol, go to **Interlocking Rule**.

5. Click on the **selection button** ... under Preview.

   The **Enter element** dialog is opened.

   The interlocking variable is visible in the lower part of the dialog under **Preview Source** and **Target**.
Initially, **Source** and **Target** are filled with the same interlocking variable.

6. Enter, under **Linking rule** and **Source (with '*' or '?')**, the name of the interlocking variable in Preview under Source.

   **Example: Interlocking 1**
   By default, under **Replace variables using Name** and **Consider interlocking variables** are selected.

7. Enter the name of the new interlocking variable under **Interlocking rule** and **Replace with**.

   **Example: Interlocking 2** The variable must have already been created beforehand.

8. In the **element entry** dialog, click on the **Process** button.

   The number of replaced connections is shown in a separate dialog.

9. Close the dialog by clicking on **OK**.

   The substituted interlocking variable of the content of the symbol is now visible under Preview and Target.

10. Close the dialog by clicking on **OK**.

11. If you want to substitute further content of the symbol with interlocking variables, carry out the described steps for this content too.
4.1.2 Example of configuration to substitute interlocking variables

Basic procedure when creating the example configuration:
1. Create an interlocking.
2. Link the symbol to an interlocking.
3. Substitute the interlocking variable.
4. Link the symbol to a screen in order to be able to use the substitution in Runtime.

CREATING AN INTERLOCKING
1. Go to Interlocking in the project tree.
2. Go to New Interlocking in the toolbar or in the context menu.
   The variable selection dialog is opened.
3. Select the desired interlocking variable with a mouse click.
4. Click on the Add button.
   The interlocking variable is added.
5. Close the dialog by clicking on OK.
   The selected interlocking variable is shown under variables.
   A new condition is added.
6. You can change the name of the interlocking if you want.
   a) To do this, left-click on the newly-created interlocking.
      The properties window of the interlocking is opened.
   b) Change the name under General and Name.
7. Right-click on Interlocking Conditions and New Interlocking Condition.
8. Click in the project properties under Condition and Logical Linking on the ... selection button.
   The formula dialog is opened.
9. Please enter an interlocking condition.
   Example: (X01.Value > 0)
10. Confirm the input by clicking on OK.

The creation of the interlocking is now complete.

LINK A SYMBOL TO THE INTERLOCKING.
1. In the project tree, go to Screens and Symbol Library.
2. Select an existing symbol that you want to link to the interlocking.
   However, you can also create a new symbol as an alternative. This procedure is described below.

3. Click on **New Symbol** to create a new symbol.
   The symbol name is created.

4. Double click on the symbol name to open a detailed view. Add a rectangle in the symbol for example.

5. Change the color of the rectangle in the properties window, under **Fill** and **Fill Color**, to any desired color.

6. Go to **Visibility/Flashing** in the properties window.

7. Under **Visibility**, click on the downward-pointing arrow of the text field to see all selection possibilities.

8. Set **From Interlocking**.

9. Select the previously-created interlocking under **Interlocking**.
   The symbol is now linked to the interlocking. The visibility of the symbol is orientated towards the formula defined in the interlocking conditions.

10. Save the changes.
    The symbol is now shown in the preview.

**SUBSTITUTION OF THE INTERLOCKING VARIABLE**

1. In the project tree, go to **Screens** and **Symbol Library**.

2. Create a new symbol.

3. Drag & drop the previously-created symbol with the interlocking to the detail view of the new symbol.
   The dialog to enter elements is opened.
   The interlocking variable is visible in the lower part of the dialog under **Preview Source** and **Target**.
   Initially, **Source** and **Target** are filled with the same interlocking variable.

4. Enter, under **Linking Rule** and **Source (with '*' or '?')**, the name of the interlocking variable in Preview under Source.

   **Example:** Interlocking 1
   
   Under **Replace variables using**, **Name** and **Consider interlocking variables** are selected by default.

5. Enter the name of the new interlocking variable under **Interlocking rule** and **Replace with**.

   **Example:** Interlocking 2 The variable must have already been created beforehand.

6. In the **element entry** dialog, click on the **Process** button.
The number of replaced connections is shown in a separate dialog.

7. Close the dialog by clicking on OK.

The substituted interlocking variable of the content of the symbol is now visible under Preview and Target.

8. Close the dialog by clicking on OK.

9. Click on the disk symbol to save the changes.

**SUBSTITUTION OF INTERLOCKING VARIABLES WHEN LINKING A SYMBOL TO A SCREEN**

You can also substitute interlocking variables if you link the corresponding symbol from the symbol library to a screen.

**Engineering:**

1. Open the desired screen to do this.

2. Select the desired symbol in the symbol library and drag & drop it into the screen.

   The dialog to enter elements (Dialog: **Enter element**) is opened.

3. Carry out the substitution as described under substitution of the interlocking variables.

**4.2 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:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 20). |
| 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 arrive at a Boolean expression, this total value has to be ORed bitwise (on page 23) with a constant. 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 the bitwise ORing example (on page 23) chapter |

**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 25) 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
# 4.2.1 List of status bits

<table>
<thead>
<tr>
<th>Bit number</th>
<th>Short term</th>
<th>Long name</th>
<th>zenon Logic long name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>M1</td>
<td>User status 1; for Command Processing: Action type &quot;Block&quot;; Service Tracking (Main.chm::/IEC850.chm::/117281.htm) of the IEC 850 driver</td>
<td>_VSB_ST_M1</td>
</tr>
<tr>
<td>1</td>
<td>M2</td>
<td>User status2</td>
<td>_VSB_ST_M2</td>
</tr>
<tr>
<td>2</td>
<td>M3</td>
<td>User status3</td>
<td>_VSB_ST_M3</td>
</tr>
<tr>
<td>3</td>
<td>M4</td>
<td>User status4</td>
<td>_VSB_ST_M4</td>
</tr>
<tr>
<td>4</td>
<td>M5</td>
<td>User status5</td>
<td>_VSB_ST_M5</td>
</tr>
<tr>
<td>5</td>
<td>M6</td>
<td>User status6</td>
<td>_VSB_ST_M6</td>
</tr>
<tr>
<td>6</td>
<td>M7</td>
<td>User status7</td>
<td>_VSB_ST_M7</td>
</tr>
<tr>
<td>7</td>
<td>M8</td>
<td>User status8</td>
<td>_VSB_ST_M8</td>
</tr>
<tr>
<td>8</td>
<td>NET_SEL</td>
<td>Select in the network</td>
<td>_VSB_SELEC</td>
</tr>
<tr>
<td>9</td>
<td>REVISION</td>
<td>Revision</td>
<td>_VSB_REV</td>
</tr>
<tr>
<td>10</td>
<td>PROGRESS</td>
<td>In operation</td>
<td>_VSB_DIREC</td>
</tr>
<tr>
<td>11</td>
<td>TIMEOUT</td>
<td>Command &quot;Timeout exceeded&quot; (command runtime exceeded)</td>
<td>_VSB_RTE</td>
</tr>
<tr>
<td>12</td>
<td>MAN_VAL</td>
<td>Manual value</td>
<td>_VSB_MVALUE</td>
</tr>
<tr>
<td>13</td>
<td>M14</td>
<td>User status14</td>
<td>_VSB_ST_14</td>
</tr>
<tr>
<td>14</td>
<td>M15</td>
<td>User status15</td>
<td>_VSB_ST_15</td>
</tr>
<tr>
<td>15</td>
<td>M16</td>
<td>User status16</td>
<td>_VSB_ST_16</td>
</tr>
<tr>
<td>16</td>
<td>GI</td>
<td>General query</td>
<td>_VSB_GR</td>
</tr>
<tr>
<td>17</td>
<td>SPONT</td>
<td>Spontaneous</td>
<td>_VSB_SPONT</td>
</tr>
<tr>
<td>18</td>
<td>INVALID</td>
<td>Invalid</td>
<td>_VSB_I_BIT</td>
</tr>
<tr>
<td>19</td>
<td>T_STD_E</td>
<td>External standard time (standard time)</td>
<td>Caution: up to version 7.50, this was the status bit T_CHG_A</td>
</tr>
<tr>
<td>20</td>
<td>OFF</td>
<td>Switched off</td>
<td>_VSB_N_UPD</td>
</tr>
<tr>
<td>21</td>
<td>T_EXTERN</td>
<td>Real time - external time stamp</td>
<td>_VSB_RT_E</td>
</tr>
<tr>
<td>22</td>
<td>T_INTERN</td>
<td>Internal time stamp</td>
<td>_VSB_RT_I</td>
</tr>
<tr>
<td>23</td>
<td>N_SORTAB</td>
<td>Not sortable</td>
<td>_VSB_NSORT</td>
</tr>
<tr>
<td>24</td>
<td>FM_TR</td>
<td>Error message transformer value</td>
<td>_VSB_DM_TR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>-------------</td>
<td>---</td>
</tr>
<tr>
<td>25</td>
<td>RM_TR</td>
<td>Working message transformer value</td>
<td>_VSB_RM_TR</td>
</tr>
<tr>
<td>26</td>
<td>INFO</td>
<td>Information for the variable</td>
<td>_VSB_INFO</td>
</tr>
<tr>
<td>27</td>
<td>ALT_VAL</td>
<td>Alternate value</td>
<td>_VSB_AVALUE</td>
</tr>
<tr>
<td>28</td>
<td>RES28</td>
<td>Reserved for internal use (alarm flashing)</td>
<td>_VSB_RES28</td>
</tr>
<tr>
<td>29</td>
<td>N_UPDATE</td>
<td>Not updated (zenon network)</td>
<td>_VSB_ACTUAL</td>
</tr>
<tr>
<td>30</td>
<td>T_STD</td>
<td>Internal standard time</td>
<td>_VSB_WINTER</td>
</tr>
<tr>
<td>31</td>
<td>RES31</td>
<td>Reserved for internal use (alarm flashing)</td>
<td>_VSB_RES31</td>
</tr>
<tr>
<td>32</td>
<td>COT0</td>
<td>Cause of transmission bit 1</td>
<td>_VSB_TCB0</td>
</tr>
<tr>
<td>33</td>
<td>COT1</td>
<td>Cause of transmission bit 2</td>
<td>_VSB_TCB1</td>
</tr>
<tr>
<td>34</td>
<td>COT2</td>
<td>Cause of transmission bit 3</td>
<td>_VSB_TCB2</td>
</tr>
<tr>
<td>35</td>
<td>COT3</td>
<td>Cause of transmission bit 4</td>
<td>_VSB_TCB3</td>
</tr>
<tr>
<td>36</td>
<td>COT4</td>
<td>Cause of transmission bit 5</td>
<td>_VSB_TCB4</td>
</tr>
<tr>
<td>37</td>
<td>COT5</td>
<td>Cause of transmission bit 6</td>
<td>_VSB_TCB5</td>
</tr>
<tr>
<td>38</td>
<td>N_CONF</td>
<td>Negative confirmation of command by device (IEC 60870 [P/N])</td>
<td>_VSB_PN_BIT</td>
</tr>
<tr>
<td>39</td>
<td>TEST</td>
<td>Test bit (IEC870 [T])</td>
<td>_VSB_T_BIT</td>
</tr>
<tr>
<td>40</td>
<td>WR_ACK</td>
<td>Writing acknowledged</td>
<td>_VSB_WR_ACK</td>
</tr>
<tr>
<td>41</td>
<td>WR_SUC</td>
<td>Writing successful</td>
<td>_VSB_WR_SUC</td>
</tr>
<tr>
<td>42</td>
<td>NORM</td>
<td>Normal status</td>
<td>_VSB_NORM</td>
</tr>
<tr>
<td>43</td>
<td>N_NORM</td>
<td>Deviation normal status</td>
<td>_VSB_ABNORM</td>
</tr>
<tr>
<td>44</td>
<td>BL_870</td>
<td>IEC 60870 Status: blocked</td>
<td>_VSB_BL_BIT</td>
</tr>
<tr>
<td>45</td>
<td>SB_870</td>
<td>IEC 60870 Status: substituted</td>
<td>_VSB_SP_BIT</td>
</tr>
<tr>
<td>46</td>
<td>NT_870</td>
<td>IEC 60870 Status: not topical</td>
<td>_VSB_NT_BIT</td>
</tr>
<tr>
<td>47</td>
<td>OV_870</td>
<td>IEC 60870 Status: overflow</td>
<td>_VSB_OV_BIT</td>
</tr>
<tr>
<td>48</td>
<td>SE_870</td>
<td>IEC 60870 Status: select</td>
<td>_VSB_SE_BIT</td>
</tr>
<tr>
<td>49</td>
<td>T_INVAL</td>
<td>External time stamp invalid</td>
<td>not defined</td>
</tr>
<tr>
<td>50</td>
<td>CB_TRIP</td>
<td>Breaker tripping detected</td>
<td>not defined</td>
</tr>
<tr>
<td>51</td>
<td>CB_TR_I</td>
<td>Breaker tripping detection inactive</td>
<td>not defined</td>
</tr>
<tr>
<td>52</td>
<td>OR_DRV</td>
<td>Value out of the valid range (IEC 61850)</td>
<td>not defined</td>
</tr>
<tr>
<td>53</td>
<td>T_UNSSYNC</td>
<td>ClockNotSynchronized (IEC 61850)</td>
<td>not defined</td>
</tr>
</tbody>
</table>
In formulas all status bits are available. For other use the availability can be limited.

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

### 4.2.2 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').

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>logical 'AND'</td>
</tr>
<tr>
<td>NOT</td>
<td>Negation</td>
</tr>
<tr>
<td>OR</td>
<td>logical 'OR'</td>
</tr>
<tr>
<td>XOR</td>
<td>logical 'EXCLUSIVE OR'</td>
</tr>
</tbody>
</table>

The operators have the following priority in the formula calculation:
### Bit Formulas

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

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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:**

\[
X01.M1 \text{ OR } X01.M2 \text{ OR } X01.M3 \text{ OR } X01.M4 \text{ OR } X01.M5 \text{ OR } X01.M6 \text{ OR } X01.M7 \text{ OR } X01.M8
\]

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 255; 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 (|), the same as in this example.

4.2.4 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.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>less</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater or equal</td>
</tr>
<tr>
<td>=</td>
<td>Equal</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>unequal</td>
</tr>
</tbody>
</table>

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 numbers are automatically converted to decimal numbers by clicking on OK. For example, 0x64 corresponds to the numerical value 100.

**Note:** The combined element is not available in the **Batch Control** module.

### 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)

### 4.2.5 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 VALUE OR STATUS OF A VARIABLE

**Example**

(X01.Value > X02.Value)

#### COMPARE COMPARISONS TO ONE OTHER ON A LOGICAL BASIS

**Example**

(X01.Value > X02.Value) AND (X01.Value = X02.Value)
5. Operating during Runtime

In order to be able to use an interlocking in Runtime, it must already have been created in full in the Editor.

Changes in Runtime are not possible.

5.1 Usage of Interlockings

Whether a Dynamic element which has been linked to an interlocking is locked or unlocked depends on the result of the logical condition. If the condition is fulfilled – the result is logical 1 –, the element is locked.

To lock a desired dynamic element, select it. In the property window Runtime/Authorization/Interlockings select one of the defined general Interlockings.
**Information**

In order to display whether the element is locked in the Runtime, you can turn on the display of a lock symbol for interlocked elements in the project properties under **User administration/Graphical identification active**.

In addition you can fashion the look of an interlocked button via property **Interlocked buttons (graphical design)** or **Locked buttons (User administration)**.