zenon
by COPA-DATA

zenon manual
Interlockings

v.8.20

Contents

1 Welcome to COPA-DATA help

ZENON VIDEO TUTORIALS

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

GENERAL HELP

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

PROJECT SUPPORT

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

LICENSES AND MODULES

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

2 Interlockings

Interlockings control the access to certain zenon objects in the 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

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

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

Symbol	Description
New interlocking	Creates a new interlocking and opens the dialog for selecting variables.
New interlocking condition	Creates a new interlocking condition.
Add variable	Opens the dialog for selecting variables.
Copy	Copies the selected condition.
Paste	Pastes the condition from the clipboard.
Delete	Deletes selected condition.
Export selected as XML...	Exports selected 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 INTERLOCKINGS

Parameter	Description
New interlocking	Creates a new interlocking and opens the dialog for selecting variables.
Paste	Pastes the interlocking from the clipboard.
Export all as XML	Exports all entries of the interlocking as an XML file.
Import XML	Imports XML files.
Help	Opens online help.

CONTEXT MENU INDIVIDUAL INTERLOCKING

Parameter	Action
Add variable	Opens the dialog for selecting a variable.
New interlocking condition	Creates a new interlocking condition.

Parameter	Action
Copy	Copies the selected interlocking.
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

Parameter	Action
Add variable	Opens the dialog for selecting variables.
Paste	Pastes the condition from the clipboard.
Help	Opens online help.

CONTEXT MENU INDIVIDUAL VARIABLE

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

CONTEXT MENU INTERLOCKING CONDITIONS

Parameter	Action
New interlocking condition	Creates a new interlocking condition.

Parameter	Action
Paste	Pastes the condition from the clipboard.
Help	Opens online help.

4 Engineering in the Editor

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

Attention

A variable must not be an interlocking variable and a result variable at the same time. This configuration would lead to an infinite loop.

The interlocking can contain one or more conditions. To do this, you must first create one or more new interlocking conditions in the **Interlocking Conditions** node.

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

The Interlocking Conditions node is also created if a new interlocking is created. The following is entered by default in each newly-created **condition**:

- **Name** property: *Condition N*
- **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. The interlocking is active in the Runtime however.

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.

Users can be informed in the Runtime if elements cannot be operated due to interlockings.

To do this, configure the **Interlocked elements** property in the project properties in the **Runtime settings**/**Runtime messages for** node.

You can find further information in the **Runtime** manual in the **Runtime messages** chapter.

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 the **Screens**/**comparison operators** (on page 24) chapter.

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. In the properties of the interlocking condition in the **Condition** group in **Logical Linking**, go to the **...** 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 OF THE RESULT VARIABLE

The result variable is initialized with the value 2 (= Interlocking is active.) when the Runtime is started.

The following values of the result variable inform you of the status of the interlocking:

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

An interlocking is also active if there is no valid value. This is the case if the variable does not contain a value, has an invalid value or the *INVALID* bit is set.

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:

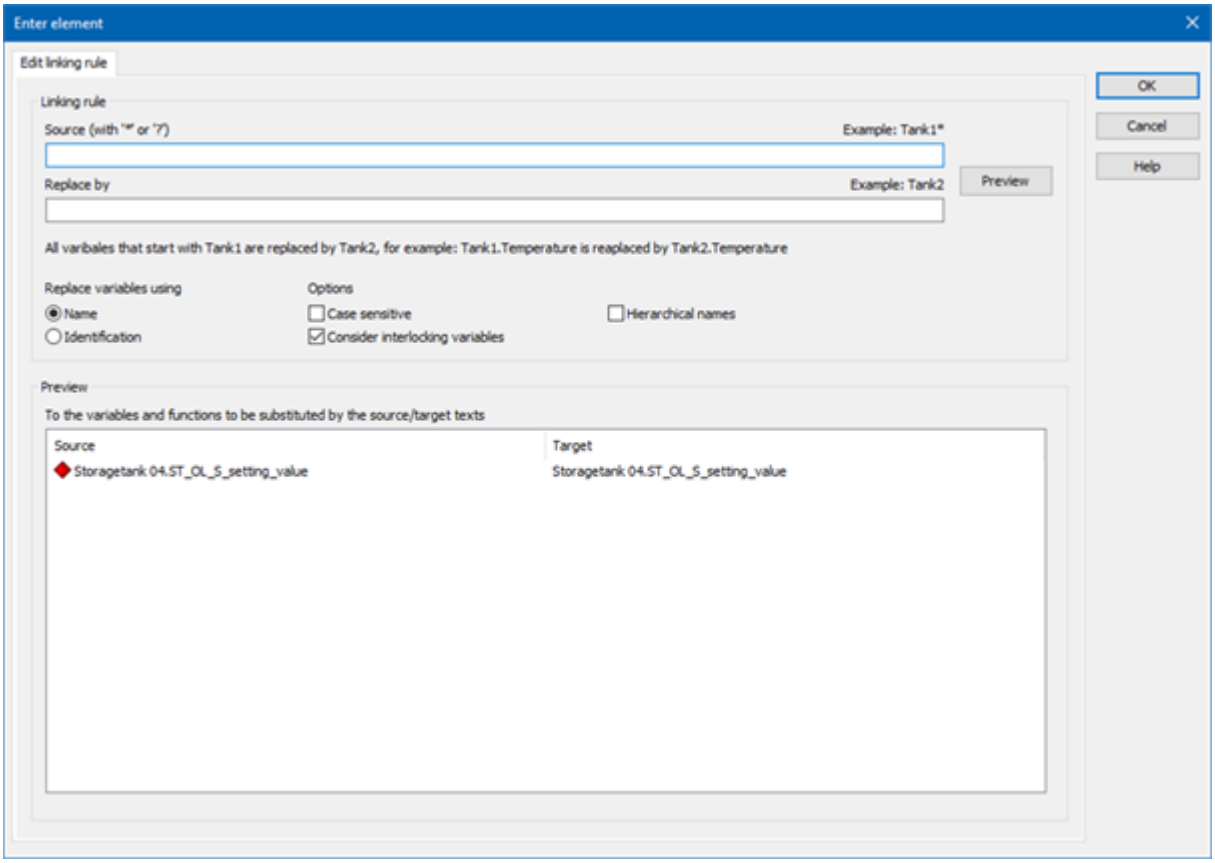
1. In the project tree under Screens, select the screen to which you want to link the symbol.
2. Open the desired screen with a double click.
3. In the project tree, go to **Screens** and **Symbol Library**.
4. Select the desired symbol with a mouse click.

   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 **element entry** 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.

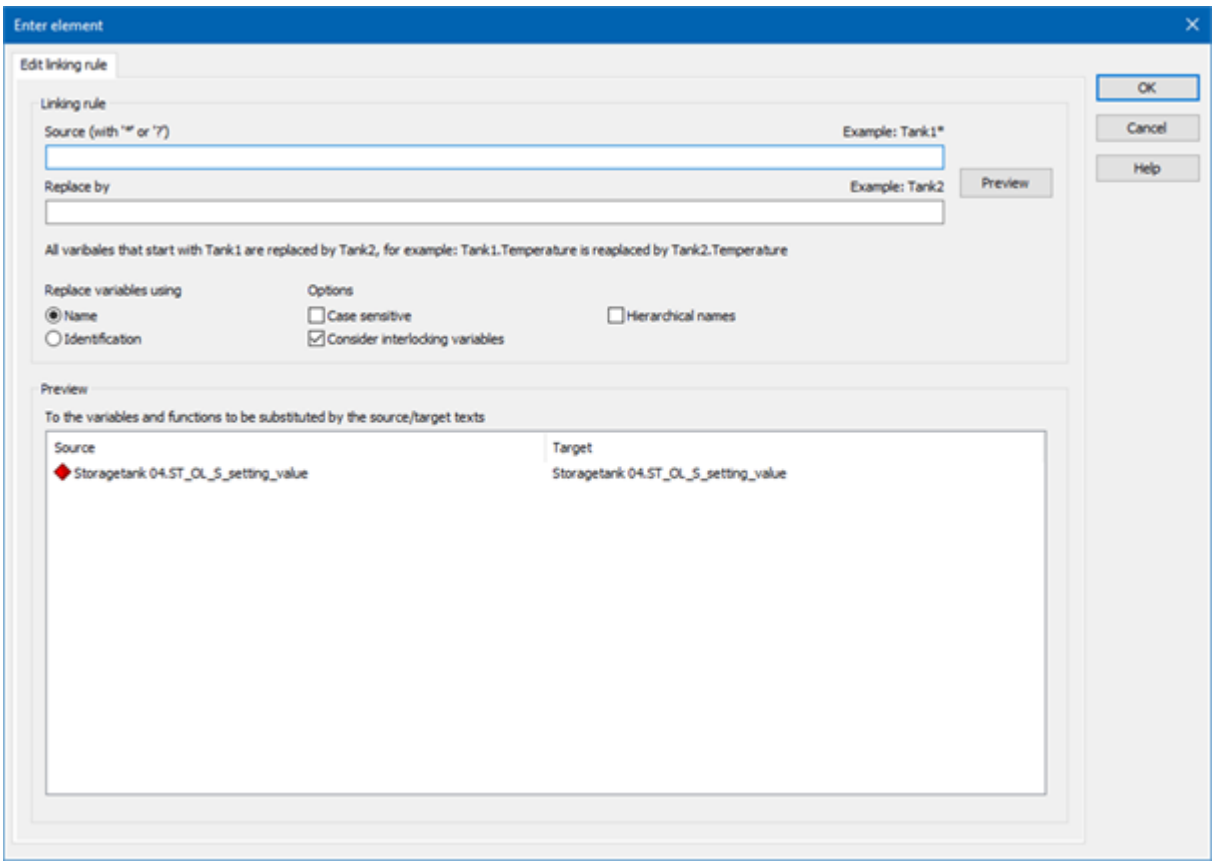
   A view of the symbol is opened.

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 **element entry** 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**.

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 the 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. In the properties of the interlocking condition in the **Condition** group in **Logical Linking**, go to the **...** 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 **element entry** dialog is opened.

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

4.2 Formula editor

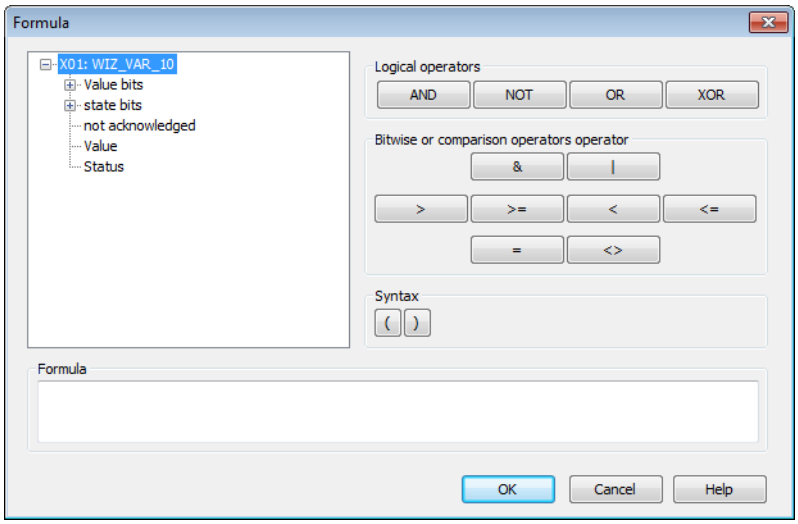
The formula editor provides support when creating formulas with logical or comparative operators with a combined element, for interlockings and command processing. If additional variables are required for a formula, create these in the **formula variables** area of the status window by clicking on the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

Parameter	Description
value bits	32 value bits (from 0 -31) are available. They describe the variable value bit by bit. For binary variables, only bit *0* is of importance, for SINT and USINT only the bits from *0-7*, etc. **Note:** The value refers to the raw value (signal range) of the variables and not to the converted measuring range.
State bits	Here you find the most commonly used status bits. You find the exact definition and use of the status bits in the Status Bits List (on page 18).
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.

Parameter	Description
	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

Bit number	Short term	Long name	zenon Logic identifier
0	M1	User status 1; for Command Processing: Action type "Block"; Service Tracking of the IEC 850 driver	_VSB_ST_M1

Bit number	Short term	Long name	zenon Logic identifier
1	M2	User status 2	_VSB_ST_M2
2	M3	User status 3	_VSB_ST_M3
3	M4	User status 4	_VSB_ST_M4
4	M5	User status 5	_VSB_ST_M5
5	M6	User status 6	_VSB_ST_M6
6	M7	User status 7	_VSB_ST_M7
7	M8	User status 8	_VSB_ST_M8
8	NET_SEL	Select in the network	_VSB_SELEC
9	REVISION	Revision	_VSB_REV
10	PROGRESS	In operation	_VSB_DIREC
11	TIMEOUT	Command "Timeout exceeded" (command runtime exceeded)	_VSB_RTE
12	MAN_VAL	Manual value	_VSB_MVALUE
13	M14	User status 14	_VSB_ST_14
14	M15	User status 15	_VSB_ST_15
15	M16	User status 16	_VSB_ST_16
16	GI	General interrogation	_VSB_GR
17	SPONT	Spontaneous	_VSB_SPONT
18	INVALID	Invalid	_VSB_I_BIT
19	T_STD_E	External standard time (standard time) **Caution:** up to version 7.50, this was the status bit T_CHG_A	_VSB_SUWI
20	OFF	Switched off	_VSB_N_UPD
21	T_EXTERN	Real time - external time stamp	_VSB_RT_E
22	T_INTERN	Internal time stamp	_VSB_RT_I

Bit number	Short term	Long name	zenon Logic identifier
23	N_SORTAB	Not sortable	_VSB_NSORT
24	FM_TR	Error message transformer value	_VSB_DM_TR
25	RM_TR	Working message transformer value	_VSB_RM_TR
26	INFO	Information for the variable	_VSB_INFO
27	ALT_VAL	Alternate value	_VSB_AVALUE
28	RES28	Reserved for internal use (alarm flashing)	_VSB_RES28
29	N_UPDATE	Not updated (zenon network)	_VSB_ACTUAL
30	T_STD	Internal standard time	_VSB_WINTER
31	RES31	Reserved for internal use (alarm flashing)	_VSB_RES31
32	COT0	Cause of transmission bit 1	_VSB_TCB0
33	COT1	Cause of transmission bit 2	_VSB_TCB1
34	COT2	Cause of transmission bit 3	_VSB_TCB2
35	COT3	Cause of transmission bit 4	_VSB_TCB3
36	COT4	Cause of transmission bit 5	_VSB_TCB4
37	COT5	Cause of transmission bit 6	_VSB_TCB5
38	N_CONF	Negative confirmation of command by device (IEC 60870 [P/N])	_VSB_PN_BIT
39	TEST	Test bit (IEC870 [T])	_VSB_T_BIT
40	WR_ACK	Writing acknowledged	_VSB_WR_ACK
41	WR_SUC	Writing successful	_VSB_WR_SUC
42	NORM	Default status	_VSB_NORM
43	N_NORM	Deviation normal status	_VSB_ABNORM

Bit number	Short term	Long name	zenon Logic identifier
44	BL_870	IEC 60870 status: *blocked*	_VSB_BL_BIT
45	SB_870	IEC 60870 status: *substituted*	_VSB_SP_BIT
46	NT_870	IEC 60870 status: *not topical*	_VSB_NT_BIT
47	OV_870	IEC 60870 status: *overflow*	_VSB_OV_BIT
48	SE_870	IEC 60870 status: *select*	_VSB_SE_BIT
49	T_INVAL	External time stamp invalid	not defined
50	CB_TRIP	Breaker tripping detected	not defined
51	CB_TR_I	Breaker tripping detection inactive	not defined
52	OR_DRV	Value out of the valid range (IEC 61850)	not defined
53	T_UNSYNC	ClockNotSynchronized (IEC 61850)	not defined
54	PR_NR	Not recorded in the Process Recorder	not defined
55	T_DEV	Configured time difference between internal and external timestamp reached.	not defined
56	RES56	reserved	not defined
57	RES57	reserved	not defined
58	RES58	reserved	not defined
59	RES59	reserved	not defined
60	RES60	reserved	not defined
61	RES61	reserved	not defined
62	RES62	reserved	not defined
63	RES63	reserved	not defined

Information

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

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

The operators have the following priority in the formula calculation:

Priority	Operator
1	& (operator for bit formulas (on page 23))
2	NOT
3	AND
4	XOR/OR

Info

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

Info

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

4.2.3 Bit formulas

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

Operator	Description
&	AND
\| | OR |

4.2.3.1 Example: ORing bitwise

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

USUAL FORMULA:

X01.M1 OR X01.M2 OR X01.M3 OR X01.M4 OR X01.M5 OR X01.M6 OR X01.M7 OR X01.M8
This query can be made much easier by the logical ORing of the overall status.

LOGICAL ORING

X01.Status & 0xFF

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

0xFF corresponds to decimal *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.

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

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

There can also be a constant to the right of the comparison operator.
These constants are entered as hexadecimal values or decimal values in the combined element. Hexadecimal 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)

COMPARE WITH VALUE BITS AND STATUS BITS

Example

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

COMPARE A VALUE WITH A DECIMAL OR HEXADECIMAL VALUE

Example

Formula: (X01.Value = 111)

Formula: (X01.Value = 0x6F)

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

Info

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

5 Operation in the Runtime:

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

Changes in the Runtime are not possible.

5.1 Usage of Interlockings

If an interlocking is linked to one of the dynamic screen elements, it is locked or unlocked depending on the logical linking in conditions of the interlocking. If the condition is fulfilled – the result is logical "*TRUE*" or *1*, the element is locked.

The dynamic element that is to be unlocked must be selected. One of the configured interlockings is selected in its properties in the **Authorization** properties group for the **Interlocking** property.

Information

In order to be able to see if the element is locked in the Runtime, the display of a padlock symbol for locked elements can be activated in the **Graphical design** properties group in the **Graphical identification active** property.

In addition, you can define the appearance of an interlocked button using the properties **Interlocked buttons** (**Graphical design** properties group) or **Locked buttons** (under **User Administration**).