



COPADATA
do it your way

zenon manual

Interlockings

v.7.20





©2015 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

1. Welcome to COPA-DATA help	4
2. Interlocking	4
3. Detail view of context menu and toolbar.....	6
4. Engineering in the Editor.....	9
4.1 Creating Interlockings	9
4.2 Formula editor	10
4.2.1 List of status bits.....	13
4.2.2 Logical operators	15
4.2.3 Bit formulas	16
4.2.4 Comparison operators.....	17
4.2.5 Examples for formulas.....	18
5. Operating during Runtime	20
5.1 Usage of Interlockings.....	20

1. Welcome to COPA-DATA help

GENERAL HELP

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

PROJECT SUPPORT

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

LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email sales@copadata.com (<mailto:sales@copadata.com>).

2. Interlocking

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.



License information

Part of the standard license of the Editor and Runtime.

INTERLOCKING OF OBJECTS

All dynamic elements except trend element and message element.

CONTEXT MENU PROJECT MANAGER

Parameters	Description
New interlocking	Creates a new interlocking and opens the dialog for selecting variables.
Export XML all	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

TOOL BAR



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 XML	Exports selected entries as an XML file.
Import XML	Imports from an XML file.
Rename	Enables the element to be renamed. It is also possible by left-clicking the field with the mouse or by pressing F2 .
Properties	Opens the property window for the selected element.
Help	Opens online help.

CONTEXT MENU INTERLOCKINGS

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

CONTEXT MENU INDIVIDUAL INTERLOCKING

Parameters	Action
Add variable	Opens the dialog for selecting a variable.
New interlocking condition	Creates a new interlocking condition.
Copy	Copies the selected interlocking.
Paste	Pastes the interlocking from the clipboard.
Delete	Deletes selected interlocking.

Export XML all	Exports all entries as an XML file.
Import XML	Imports from an XML file.
Rename	Enables the element to be renamed. It is also possible by left-clicking the field with the mouse or by pressing F2 .
Properties	Opens the property window for the selected element.
Help	Opens online help.

CONTEXT MENU GROUP VARIABLES

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

CONTEXT MENU INDIVIDUAL VARIABLE

Parameters	Action
Delete 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

Parameters	Action
New interlocking condition	Creates a new interlocking condition.
Paste	Pastes the condition from the clipboard.
Help	Opens online help.

4. Engineering in the Editor

4.1 Creating Interlockings

Select **Interlockings** in the Project Manager to display the defined interlockings in the Detailview of the Project Manager. A right click on **General Interlockings** opens a popup menu, in which a new interlocking can be created.

The new Interlocking can be renamed in the properties window. A right click on **Variables** again opens a popup menu. Here, one or more variables can be linked to the according interlocking.

Attention

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

The interlocking can be linked to a condition. To do this, one or more conditions have to be created in the properties window. For each condition, a binary formula can be entered in the properties window under **Logical link**.

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 10) can be used to define binary and numeric formulas as a locking condition. It is opened by clicking on the **Logical linking** 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 17).

Information

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

4.2 Formula editor

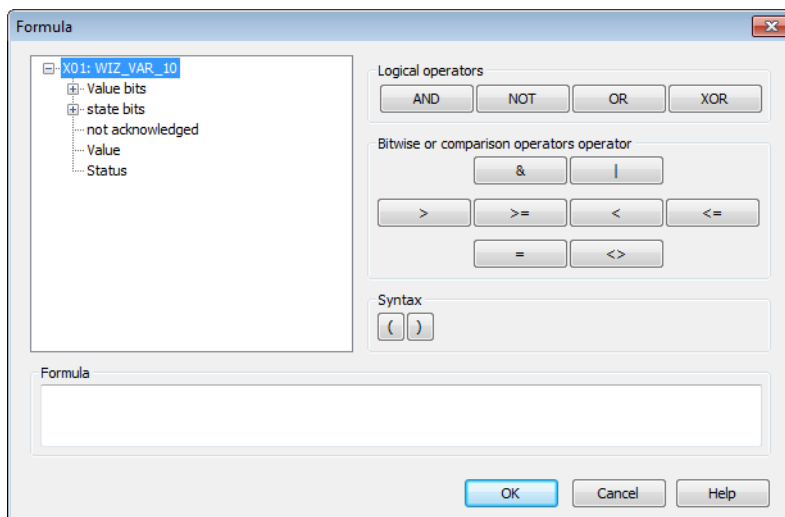
The formula editor provides support when creating formulas with logical or comparative operators with a combined element, for interlockings and commands. If additional variables are required for a formula, create these in the **formula variables** area of the status window by clicking on the **Add**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

Note on the input of decimal points:

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

CREATING A FORMULA

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



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

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

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



Information

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

THE MEANING OF THE BITS:

Parameters	Description
value bits	32 value bits (from 0 -31) are available. They describe the variable value bit by bit. For binary variables, only bit 0 is of importance, for SINT and USINT only the bits from 0-7, etc. Note: The value refers to the raw value (signal range) of the variables and not to the converted measuring range.
status 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 13).
not acknowledged	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	All values (value bits and status bits) in the formula are considered as binary value and can be linked with logical operators such as AND or OR. The total value and overall status are an exception to this. In order to get a Boolean result this total value has to be ORed with a constant <code>bitwise</code> (on page 16). For this, we use the operator <code>&</code> . For the result 0 (false) of this logical ORing we get the binary value 0 (false), otherwise 1 (true). Example: see chapter Example bit by bit ORing (on page 16)



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 18) 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 label
0	M1	User status 1	_VSB_ST_M1
1	M2	User status 2	_VSB_ST_M2
2	M3	User status 3	_VSB_ST_M3
3	M4	User status 4	_VSB_ST_M4
4	M5	User status 5	_VSB_ST_M5
5	M6	User status 6	_VSB_ST_M6
6	M7	User status 7	_VSB_ST_M7
7	M8	User status 8	_VSB_ST_M8
8	NET_SEL	Select in the network	_VSB_SELEC
9	REVISION	Revision	_VSB_REV
10	PROGRESS	In operation	_VSB_DIREC
11	TIMEOUT	Runtime exceedance	_VSB_RTE
12	MAN_VAL	Manual value	_VSB_MVALUE
13	M14	User status 14	_VSB_ST_14
14	M15	User status 15	_VSB_ST_15
15	M16	User status 16	_VSB_ST_16
16	GI	General interrogation	_VSB_GR
17	SPONT	Spontaneous	_VSB_SPONT
18	INVALID	Invalid	_VSB_I_BIT
19	T_CHG_A	Daylight saving time/winter time announcement	_VSB_SUWI
20	OFF	Switched off	_VSB_N_UPD
21	T_EXTERN	Real time external	_VSB_RT_E
22	T_INTERN	Realtime internal	_VSB_RT_I
23	N_SORTAB	Not sortable	_VSB_NSORT
24	FM_TR	Error message transformer value	_VSB_DM_TR
25	RM_TR	Working message transformer value	_VSB_RM_TR
26	INFO	Information for the variable	_VSB_INFO
27	ALT_VAL	Alternate value If no value was transferred, the defined alternate value is used	_VSB_AVALUE

		otherwise the last valid value is used.	
28	RES28	Reserved for internal use (alarm flashing)	_VSB_RES28
29	N_UPDATE	Not updated	_VSB_ACTUAL
30	T_STD	Standard time	_VSB_WINTER
31	RES31	Reserved for internal use (alarm flashing)	_VSB_RES31
32	COT0	Cause of transmission bit 1	_VSB_TCB0
33	COT1	Cause of transmission bit 2	_VSB_TCB1
34	COT2	Cause of transmission bit 3	_VSB_TCB2
35	COT3	Cause of transmission bit 4	_VSB_TCB3
36	COT4	Cause of transmission bit 5	_VSB_TCB4
37	COT5	Cause of transmission bit 6	_VSB_TCB5
38	N_CONF	Negative acceptance of Select by device (IEC 60870)	_VSB_PN_BIT
39	TEST	Test bit (IEC870 [T])	_VSB_T_BIT
40	WR_ACK	Writing acknowledged	_VSB_WR_ACK
41	WR_SUC	Writing successful	_VSB_WR_SUC
42	NORM	Normal status	_VSB_NORM
43	N_NORM	Deviation normal status	_VSB_ABNORM
44	BL_870	IEC 60870 Status: blocked	_VSB_BL_BIT
45	SB_870	IEC 60870 Status: substituted	_VSB_SP_BIT
46	NT_870	IEC 60870 Status: not topical	_VSB_NT_BIT
47	OV_870	IEC 60870 Status: overflow	_VSB_OV_BIT
48	SE_870	IEC 60870 Status: select	_VSB_SE_BIT
49	T_INVALID	Time invalid	not defined
50	CB_TRIP	Breaker tripping detected	not defined
51	CB_TR_I	Breaker tripping detection inactive	not defined
52	RES52	reserved	not defined
53	RES53	reserved	not defined
54	RES54	reserved	not defined
55	RES55	reserved	not defined
56	RES56	reserved	not defined
57	RES57	reserved	not defined

58	RES58	reserved	not defined
59	RES59	reserved	not defined
60	RES60	reserved	not defined
61	RES61	reserved	not defined
62	RES62	reserved	not defined
63	RES63	reserved	not defined



Information

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

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

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 16))
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

Example: ORing bitwise

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

USUAL FORMULA:

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

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

LOGICAL ORING:

`X01.Status & 0xFF`

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

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

If, for example, all user status bits except the user status bit M7 should be queried, the binary statement for this would be: 10111111. Bit 7 is not of interest and is thus set to 0. This corresponds to 0xBF in hexadecimal. The expression for the formula is then: `x01.Status & 0xBF`.

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

Example:

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

```
x01.value = 202
```

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

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

4.2.4 Comparison operators

Comparison operators serve 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 then 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. (the constants can only be integers; a comparison to a floating point number is not possible.)

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

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



Example

X01.value >= X02.value

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

X01.value = 0x64

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

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

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

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 ANALOGUE VALUE OR STATUS OF A VARIABLE

Example

(X01.Value > X02.Value)

COMPARE ANALOG VALUES WITH EACH OTHER ON A LOGICAL BASIS

Example

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

COMPARE WITH VALUE BITS AND STATUS BITS

Example

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

COMPARE A VALUE WITH A DECIMAL OR HEXADECIMAL VALUE

Example

Formula: (X01.Value = 111)

Formula: (X01.Value = 0x6F)

If a hexadecimal values is used, this is later transferred to decimal by clicking on α . 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. Operating during Runtime

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).*