COPA-DATA
do it your way

zenon manual

Command Sequencer

v.7.20

Contents

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. Command Sequencer

The Command Sequencer module allows commands from the Command Processing module to be compiled into processes in zenon, to visualize these and to execute user interactions if required.

3. Introduction

The module Command Sequencer consists of three parts:

1. The engineering environment in the zenon Editor:
   Here, the data for command sequences is applied from the configuration in the Command Processing module.
2. The Command Sequencer Editor in zenon Runtime:
   With this Editor, the command sequences are created in zenon Runtime. The configured command processing is the basis for command sequences. During the process, the respective status of the command processing is displayed in the Command sequencer Editor and you can make changes to the command sequence process.

PARTICULAR ASPECTS OF THE COMMAND SEQUENCER MODULE

In contrast to most other zenon modules, a large part of the project configuration, namely the creation of a command sequence, is done in Runtime and not in the zenon Editor. This entails special features which are dealt with in the respective chapter.

The module is designed in a way which makes it completely independent of the control. This means that the data communication take place via all available zenon energy drivers with any PLCs or even IED. They only execute the process actions. The complete editing of a command sequence is carried out in the computer in the Command Sequences Editor. No modification to the PLC code is necessary when a change is made to a command sequence.

PRINCIPLE STRUCTURE OF THE COMMUNICATION

- Command variable:
  The command variable is the variable that is linked during project configuration in the Command Processing module for the respective command action.
  With this variable, set values are transferred to the PLC when a step is executed.
- Response variable:
  This variable is used to read back values from the PLC for evaluations.

SCHEMA:

The PLC communicates with the zenon Energy driver, which in turn communicates with the command processing in zenon Runtime. The command processing sends the values back to the command sequence editor, where they are processed. Whilst executing a command sequence, the command sequence editor works synchronously to zenon Runtime in a cycle of 100 ms.

3.1 block - supported action types

In order to be able to use configuration of the command processing in the Command Sequencer module, at least one action must be configured in the Command Sequencer module.

The following action types of command processing are supported in the Command Sequencer module:

- Command
- Forced command
- Direct set value input
- Direct status input
- Replace
- Revision
- Direct correction
- Block
- Release
- Check response value
- Lock

> **Information**
>
> *You can find further information in the Energy Edition manual in the Action types chapter.*

3.2 Variables for command sequences

zenon When compiling the command processing, the data model for command sequences is also created.

This is created as follows:

- An item of switchgear is created for each response variable, which gets the variable name or variable identification of the response variable as a name.
  This created switching device is assigned the response variable. Return TAG is used as a type.
  Data type is: `Numerical`.
  The naming of the switching device can be configured in the **Command Sequencer** project property in the **Display name in Command Sequencer Editor** property.
- A step is created for each action that is configured in the Command Processing module and for which a response variable is available. The step is given the name of the action that is displayed in the command processing tree.
- The response variable is linked to each step created as a parameter. These parameters are, for example, significant when creating transitions (on page 36).

DISPLAY NAME IN COMMAND SEQUENCER EDITOR

The text display of a variable in Runtime is configured in the **Command Sequencer** project property group in the **Display name in Command Sequencer Editor** property.

Depending on the configuration, the variable is displayed in the command sequence grid accordingly when shown in Runtime.

Possible display names:

- Variable name
- Variable identification
- symbolic address

VALIDATION:

A check is carried out when compiling the Runtime files in the zenon Editor.

In doing so, a check is made to see if the naming of the created switching device is unique. If this is not unique, an error message is issued in the output window of the Editor. No objects that are available for the command sequences are created for the response variable.

ERROR MESSAGES:

```
'<VariablenameX>' variable ignored for the command sequence data model, because
the '<VariablennameY>' variable has already created an entry '<Naming in Command
Sequences>'! Possible cause: not a unique ID or symbolic address.
```

A check is also carried out to see whether the variable can provide an invalid object name for the command sequence object.

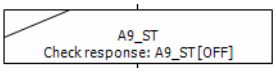
```
'<Variablename>' variable ignored for the command sequence data model because this
gives and invalid '<command sequence object name' object name!
Possible Reason: empty Identification or empty symbolic Address.
```

3.2.1 Examples of views - Display name in command sequence grid

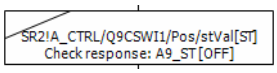
Examples of views for possible configurations of the variable names.

DISPLAY NAME IN COMMAND SEQUENCE GRID

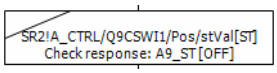
VARIABLE NAME



VARIABLE IDENTIFICATION



SYMBOLIC ADDRESS



4. Terminology

The following terms are used In the zenon Command Sequencer module:

Parameters	Description
Switchgear	Physically available element, for example: Switch or disconnector.
Command Sequencer Editor	Part of the Command Sequencer module to control the process of command processing. The command sequence editor carries out a command sequence in test mode. The complete process of the command sequence can be configured in the command sequence and the command sequences can be administered in the editing mode.
Command sequence	Command steps are compiled and saved in command sequences. These steps are then transferred to the controller by the control system.
Begin parallel branch	Element that ensures the breakdown of the command sequence process into two or more branches.
Transition	Element of the Command Sequencer module that contains a condition. The element is used after a step to ensure a defined transition from one step to the next.
Step	Execution of an action from the command processing, such as: `Switching command: OFF.`
End parallel branch	Element that combines the separation of the command sequence process into two or more branches back into one branch.
Branch	Area of the Command Sequencer module that allows separation into two or more branches, of which only one branch can be active during the process. It is an either/or branch. A branch always starts with the **`Begin branch`** and ends with the **`End branch`** element.
Assignment of switching device	Element of the Command Sequencer module that instigates the assignment of a switching device in Runtime: ▶ With this element, several (or all) response variables can be reserved (assigned) in advance. The NET_SEL status bit is set for this reservation. ▶ Unlocking is also carried out using a switchgear assignment element. ▶ After the command sequence process, all NET_SEL status bits are automatically deleted again. You can find further information on the status bit in the Status processing manual in the Select in the

	network (NET_SEL) section.
Branch	An execution area in the Command Sequencer module. Steps, transitions and jump targets can be placed on it.
End of element	Element of the Command Sequencer module with which each command sequence is forced to end.
Jump target	Element of the Command Sequencer module that allows a direct jump to a defined location of a branch.

5. Procedure

Configuration and use use of the Command Sequencer module takes place in three main steps:

1. Configuration of the command processing in the zenon Editor.
2. Creation of the command sequences in the command sequence editor (on page 15) in Runtime.
3. Execution of the command sequence in Runtime.

CREATION OF A COMMAND SEQUENCE

The user creates a command sequence in zenon Runtime. The selectable steps that correspond to the actions of command processing serve as a basis for this configuration. To do this, the command sequence in Runtime must be in edit mode.

EXECUTION OF A COMMAND SEQUENCE

The user executes the command sequence in Runtime. To do this, they first change the command sequence mode to test mode. The command sequence is then started.

The user can no longer alter command sequences in test mode. The command sequence must be switched to edit mode again in order to edit it.

6. Licensing

The Command Sequencer module offers you the possibility to create, execute and configure command processes in a graphic flow chart.

The module can only be licensed in addition to the Energy Edition.

License information

Must be licensed for Editor and Runtime (single-user, Server, Standby and Client).

If both the Batch Control module and the Command Sequencer (on page 5) module, which both require a license, are licensed at the same time, selection of the module used is carried out by means of the project setting.

To select the preferred module in Runtime:

- Click on the node of your project in the Editor.
- Go to the **Runtime settings** project properties group.
- Select, for **License module in Runtime** of the the **Preferred module** property, `command sequence`(default) or `Batch Control`.

The selected model is then available in Runtime for further project configuration.

7. Engineering in the Editor

To be able to use the Command Sequencer module in Runtime, you must first do the following in the zenon Editor:

- Configure command processing.
- Configure variables (on page 7).
- Create a `command sequence` screen (on page 12)
- In the `command sequence` screen, add the `command sequence editor` control element.
- Create a screen switch function (on page 13) for the `command sequence` screen

If the configuration in zenon Editor changes, this is applied by compiling the Runtime files and reloading them in zenon in the Command Sequencer module.

7.1 Create a command sequence screen

In a command sequence screen, command sequences are:

- created
- managed
- executed in the Runtime

The display size of the screen depends on the selected template size.

To create a command sequence screen:

1. Select the **New screen** command in the **Screens** node
2. In the **screen type** column, select command sequence in the combo box
3. in the menu bar **Control elements** select one of the three default settings:
   - Insert template ...: Offers templates for different resolutions and the selection of command sequences.
   - Command sequence editor: adds the screen element for the command sequence editor.
4. the default elements for the desired setting are created
5. if needed you can add additional control elements from menu **Control elements**
6. create a screen switch in order to display the screen in the Runtime
7. Configure the display in Runtime (for details, see the **screens** manual, **Designing lists** chapter)

7.2 Command sequence screen switching

To use command sequences in Runtime, configure a screen switch function to a command sequence screen:

1. Select the **New function** command in the **Functions** node.
2. select the **Screen switching** function.
3. select the Command Sequencer screen

Link the function with a button on the screen in order to be able to switch in Runtime.

Information

The configuration of the Command processing module serves as a basis for operation in the Command Sequencer screen.

8. Project backup for command sequences

a project backup in the zenon Editor does not take into account the command sequences configured in Runtime.

Back up the corresponding Runtime files manually.

BACK UP THE COMMAND SEQUENCE RUNTIME FILES:

1. Switch to the folder of the Runtime files.
   `C:\Users\Public\Documents\zenon_Projects\[Workspacename]\[Project name]`

   Note: This folder is only available if the project has been compiled at least once in zenon .
   Tip: Highlight the desired project in the zenon Editor and press the keyboard shortcut `Ctrl + Alt + R` to go directly to the Runtime folder.
2. Copy the `Sequences` folder.
3. Add the `Sequences` folder to the Runtime folder of the project backup again.
   Note: Runtime should be closed during the copy process and then reopened again.

9. Function authorizations

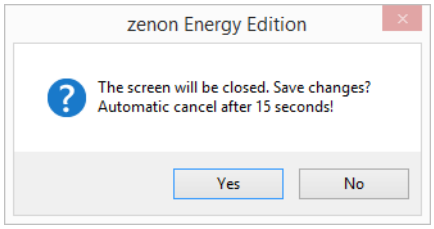
There are no separate function authorizations for the Command Sequencer module.

10. Command sequences in Runtime

All administration, creation and processing of command sequences is carried out in Runtime. Editing in the zenon Editor is not possible.

SAVING OF COMMAND SEQUENCES WHEN CLOSING RUNTIME

If Runtime is closed and there are still command sequences that have not been saved, you are asked if these command sequences are to be saved. In order for this query to not prevent Runtime closing, Runtime is automatically closed after 15 seconds if nothing is entered. Unsaved command sequences are then discarded.



- Clicking on the Yes button saves all changes for all command sequences open in the command sequences Editor. Runtime is then closed.
- Clicking on the No button closes Runtime without saving changes to the command sequences.

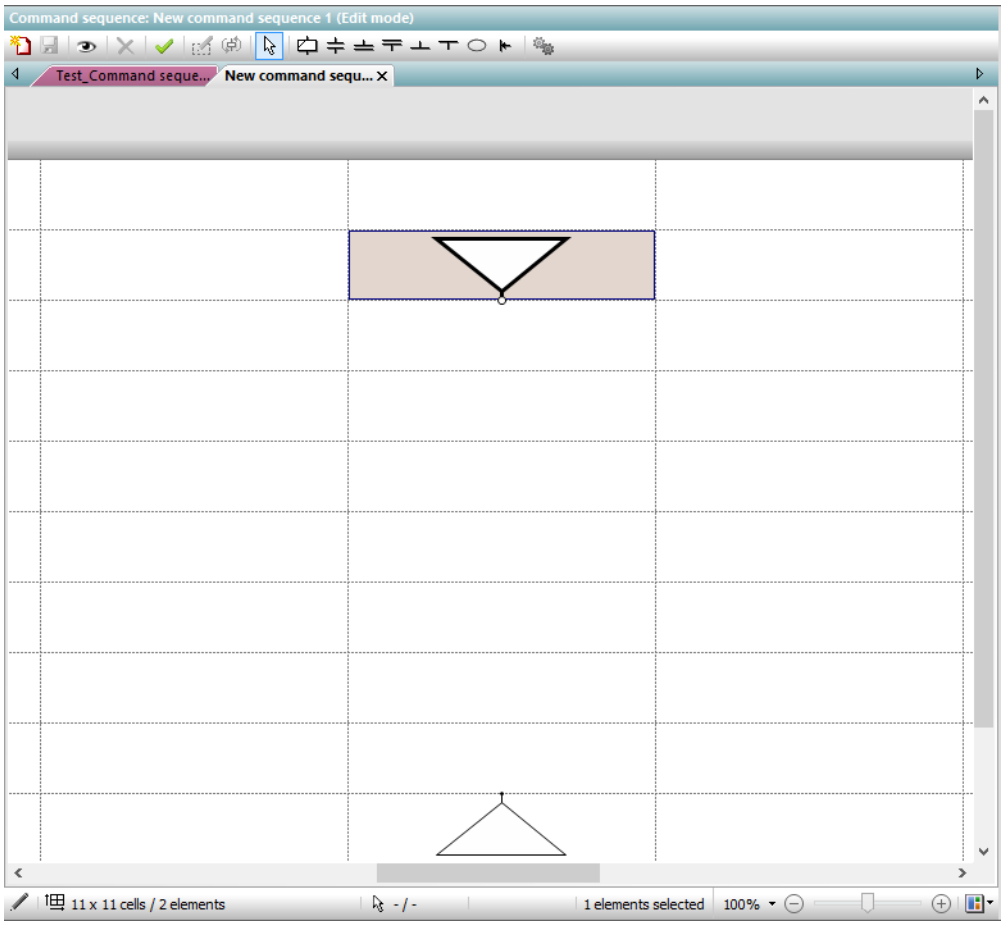
10.1 Command Sequencer Editor

The command sequence editor is the graphical user interface for the configuration of command sequences in Runtime.

The following are available for the command sequence editor:

- Tool bars
  The design of the tool bars depends on the command sequence editor mode (on page 28).
- Tabs
- Command sequence grid (on page 27)
  project configuration area

- Dockable windows



Parameters	Description
Header of the editor	Information about: ▶ Current command sequence ▶ Mode
command sequence editor menu bar	Menu bar (on page 29) with symbols to configure a command sequence.
Tabs with opened command sequences	Select the open command sequences by clicking on the respective tab. In test mode, the color of the tab corresponds to the color of the execution status (on page 62) of the command sequence.
Command sequence grid	Diagram of the project configuration. Configuration by dragging & dropping the elements from the menu bar. Hint: To enlarge or reduce the grid, position the mouse pointer at the edge and drag it in the desired direction with the mouse button held down.
Mode display	Status display of the selected mode: ▶ Edit mode: • Symbol for editing mode (pencil) • Size of the command sequence window • Number of configured elements ▶ Test mode: • Symbol for test mode: - Automatic - Manual • Symbol for the status (green thumb: ready to start) • Execution step: - in execution - completed
Cursor position	Shows line and column of the selected choice in the command sequence grid.
Zoom bar	Setting of the zoom factor for the command sequence grid with slider.
+	Enlarges the zoom factor by 25% per click.
-	Reduces the zoom factor by 25% per click.

Selection of the dockable windows	Opens drop-down list to select the dockable windows: ▶ List of command sequences (on page 19) Selected dockable windows are shown or hidden.

10.1.1 Context menu - tabs with opened command sequences

If several command sequences are open in the command sequence editor, these are represented with tabs. The tabs represent the command sequences that are open in the command sequences editor. The configuration of the command sequence in the command sequence grid is shown by clicking on a tab.

TAB CONTEXT MENU

Parameters	Description
Save	Saves the current command sequence
Close	Closes the current command sequence
Close all others	Closes all open tabs/command sequences with the exception of the one that is currently selected.
Group horizontally	Shows all open tabs in a new view. The view opens in a new window under the current view: ▶ Move view Opens tab in a new window. ▶ Open view at the same time Opens selected tab in a new window and leaves it in the tab list.
Group vertically	Shows all open tabs in a new view. The view opens in a new window next to the current view: ▶ Move view Opens tab in a new window. ▶ Open view at the same time Opens selected tab in a new window and leaves it in the tab list.

Command sequences can be displayed and opened in two groups next to each other or underneath each other. To open a command sequence in a new group:

1. Select, in the context menu of the command sequence, the **Group horizontally** or **Group vertically** command

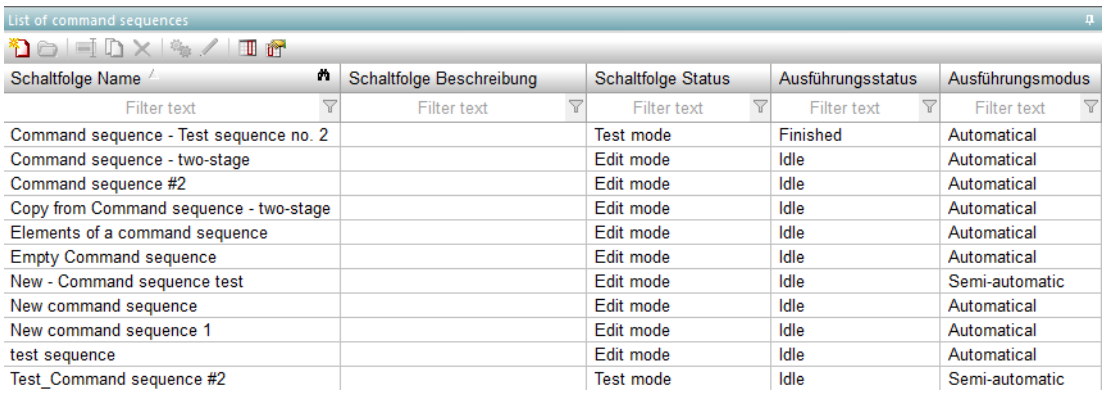
- Select the type of display:
  - Move display
  - Open display at the same time

The control elements are always only applicable to the active command sequence of the active group.

- The active group is emphasized in color.
- The active tab is emphasized with bold font. Information on the active tab is shown in the title bar of the editor.
- Tabs can be moved and arranged by dragging & dropping, including between groups.
  Hint: Use this possibility of moving to return to the view with one group when two groups are open.

10.1.2 Dockable windows - list of command sequences

The list of command sequences lists all configured command sequences. Command sequences that have already been configured are edited and administered in the list of command sequences and new command sequences are created.



Schaltfolge Name	Schaltfolge Beschreibung	Schaltfolge Status	Ausführungsstatus	Ausführungsmodus
Filter text	Filter text	Filter text	Filter text	Filter text
Command sequence - Test sequence no. 2		Test mode	Finished	Automatical
Command sequence - two-stage		Edit mode	Idle	Automatical
Command sequence #2		Edit mode	Idle	Automatical
Copy from Command sequence - two-stage		Edit mode	Idle	Automatical
Elements of a command sequence		Edit mode	Idle	Automatical
Empty Command sequence		Edit mode	Idle	Automatical
New - Command sequence test		Edit mode	Idle	Semi-automatic
New command sequence		Edit mode	Idle	Automatical
New command sequence 1		Edit mode	Idle	Automatical
test sequence		Edit mode	Idle	Automatical
Test_Command sequence #2		Test mode	Idle	Semi-automatic

LIST OF COMMAND SEQUENCES - OVERVIEW:

- New command sequences are created.
- Command sequences that have already been configured in the command sequence editor are loaded for further editing.
- Command sequences are renamed.
- Command sequences are duplicated.
- Command sequences are deleted.
- Command sequences are switched to test mode.

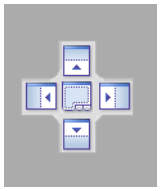
- Command sequences are switched to edit mode.
- Columns are selected and formatted.

The columns of the list for command sequences can be sorted and filtered. The columns can be moved by means of drag&drop. Columns can be shown and hidden with the context menu (right mouse click).

Selection and positioning

POSITIONING AIDS

When moving windows from the Editor interface, positioning aids are displayed. These represent windows or their borders.



This element represents a window area in the Editor.

This element represents the border area of the Editor.

POSITION WINDOW

To position an element as docked:

1. Move the element with the mouse into the desired area
2. The positioning aid is displayed
3. This represents a window and its areas:
   a) Center: whole window
   b) Top: upper half
   c) Bottom: lower half
   d) Right: right half
   e) Left: left half

   or the border of the Editor
4. Move the mouse to the central positioning aid or to a positioning aid on the border of the editor and from there to the desired area

5. The area in the Editor where the element was placed when the mouse button was released is colored in blue
6. Move the mouse within the positioning aid to the desired area that is displayed in blue
7. Let the mouse button go and the element is placed

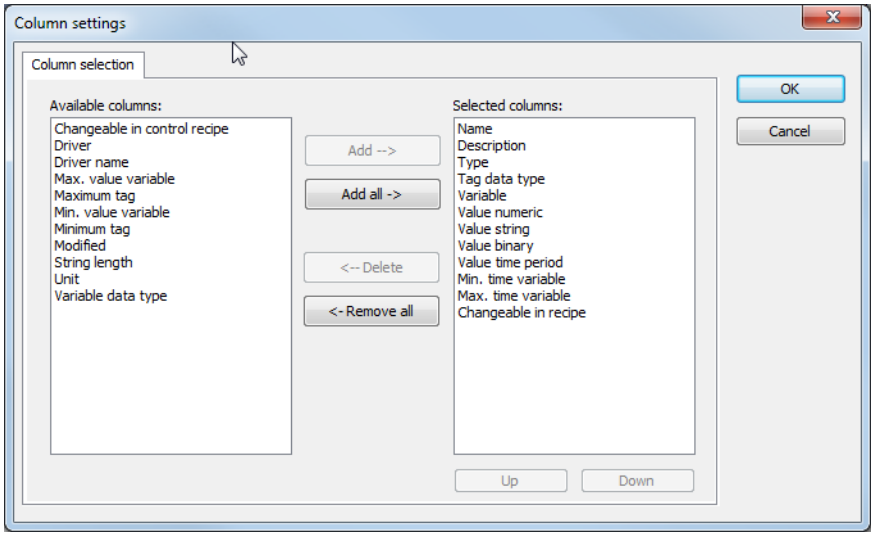
If a window is placed on a pre-existing window, both windows are displayed at the same location using tabs.

Tool bar - list of command sequences

Parameters	Description
New command sequence	Creates new empty command sequence
Open command sequence in Editor	Opens the selected command sequence in the command sequence editor. Note: Not active if no command sequence is selected.
Rename command sequence	Opens dialog to create a new command sequence or rename a command sequence. Note: Not active if no command sequence is selected. Note: Not active if the selected command sequence is in test mode.
Duplicate command sequence	Duplicates selected command sequence and opens dialog to create a new command sequence. Note: Not active if no command sequence is selected. Note: duplicated command sequence is automatically created in edit mode. When duplicating command sequences, the existing name is supplemented with the prefix "**Copy of**". If the maximum length is exceeded by this, the name is shortened to the allowed length starting from the last character.
Delete command sequence	Deletes selected command sequence(s). Multiple selection is possible. A command sequence that is executed cannot be deleted. An information dialog opens in this case. Note: Not active if no command sequence is selected. Note: Before final deletion, an additional dialog appears requesting confirmation of whether the selected command sequence(s) are really to be deleted for good.
Switch command sequence to test mode	Switches selected command sequence(s) to test mode (on page 58). A validation of the configured command sequence is carried out automatically. Test mode only starts if the command sequence has been configured without errors. Otherwise a notice dialog appears informing you that the command sequence is invalid. The command sequence cannot be started. Note: Not active if no command sequence is selected.

Switch command sequence to edit mode	Switches a command sequence that is currently running in test mode back into edit mode (on page 29) in order to make changes. Note: Not active if no command sequence is selected.
Column selection	Opens a dialog (on page 23) to select columns that are to be displayed.
Column formats	Opens dialog (on page 24) for the configuration of text and background colors for the display of columns in the list view.

Column selection

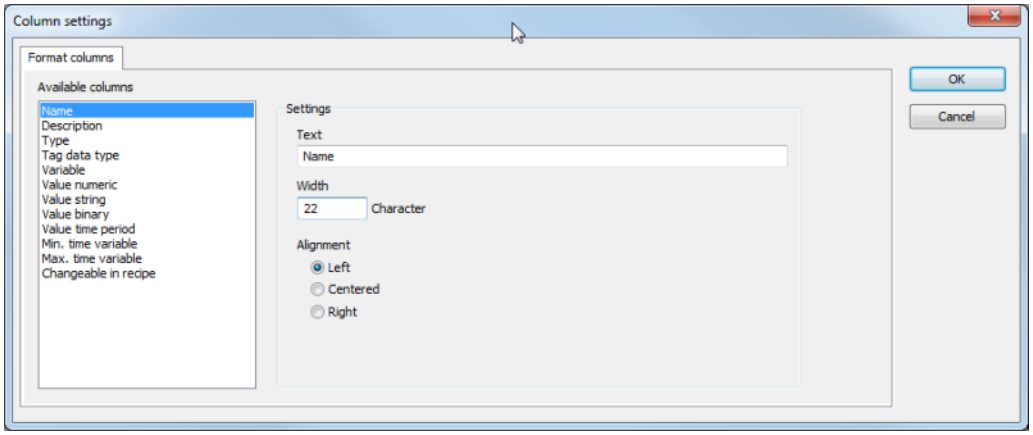


Button	Function
Available columns	List of columns that can be displayed in the table.
Selected columns	Columns that are displayed in the table.
Add	Moves the selected column from the available ones to the selected items. After you confirm the dialog with OK, they are shown in the detail view.
Add all	Moves all available columns to the selected columns.
Remove	Removes the marked columns from the selected items and shows them in the list of available columns. After you confirm the dialog with OK, they are removed from the detail view.
Remove all	All columns are removed from the list of the selected columns.
Up	Moves the selected entry upward. This function is only available for unique entries, multiple selection is not possible.
Down	Moves the selected entry downward. This function is only available for unique entries, multiple selection is not possible.

CLOSE DIALOG

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

Column Format



Parameters	Description
Available columns	List of the available columns via **Column selection**. The column selected here is configured using the settings in the **Parameters** section.
Parameters	Settings for selected column.
Labeling	Name for column title. The column title is online language switchable. For this you must enter the @ character in front of the name.
Width	Width of the column in characters. Calculation: Number time average character width of the selected font.
Alignment	Alignment. Possible settings: ▸ Left: Text is justified on the left edge of the column. ▸ Centered: Text is displayed centered in the column. ▸ Right: Text is justified on the right edge of the column.
OK	Applies settings and closes the dialog.
Cancel	Discards settings and closes the dialog.

Tool bar - list of command sequences

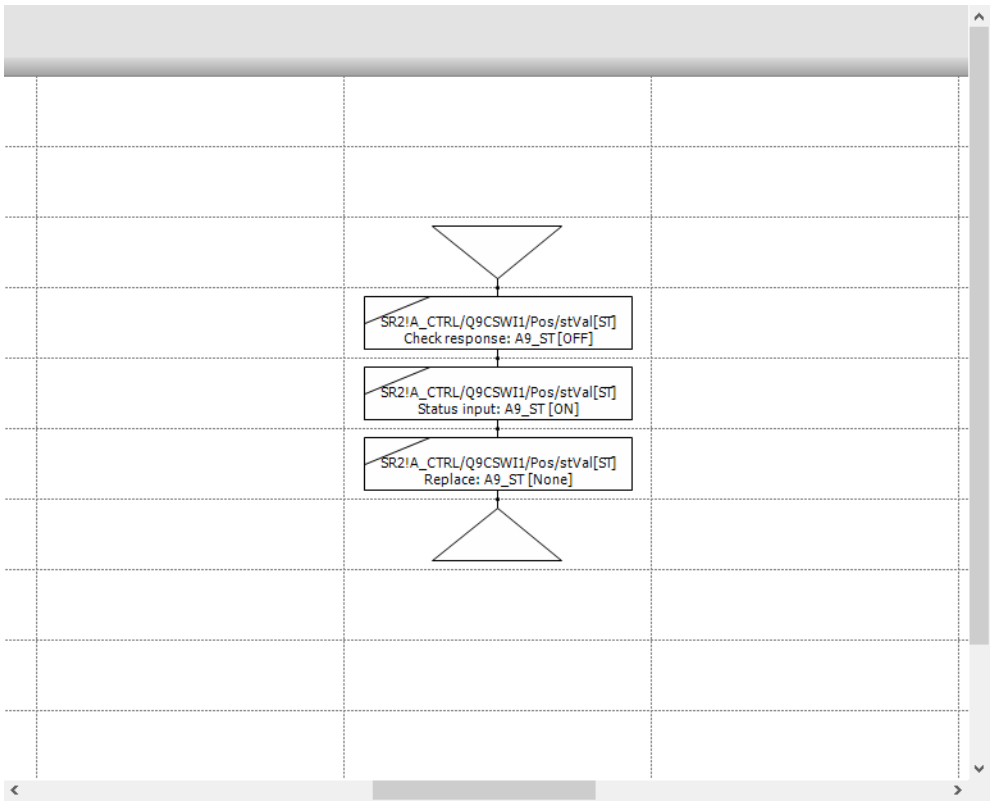
Parameters	Description
New command sequence	Creates an empty command sequence and opens the dialog to create a new command sequence.
Open in command sequence editor	Opens the selected command sequence in the command sequence editor (on page 29). Note: Not active if no command sequence is selected.
Rename	Opens dialog to create a new command sequence or rename a command sequence. Note: Not active if no command sequence is selected. Note: Not active if the selected command sequence is in test mode.
Duplicate	Duplicates selected command sequence and opens dialog to create a new command sequence. Note: Not active if no command sequence is selected. Note: duplicated command sequence is automatically created in edit mode. When duplicating command sequences, the existing name is supplemented with the prefix "**Copy of**". If the maximum length is exceeded by this, the name is shortened to the allowed length starting from the last character.
Delete	Deletes selected command sequence(s). Multiple selection is possible. Note: Not active if no command sequence is selected. Note: Before final deletion, an additional dialog appears requesting confirmation of whether the selected command sequence(s) are really to be deleted for good.
Switch to test mode	Switches selected command sequence(s) to test mode (on page 58). A validation of the configured command sequence is carried out automatically. Test mode only starts if the command sequence has been configured without errors. Otherwise a notice dialog appears informing you that the command sequence is invalid. The command sequence cannot be started. Note: Not active if no command sequence is selected.

Switch to edit mode	Switches a command sequence that is currently running in test mode back into edit mode in order to make changes. Note: Not active if no command sequence is selected.

10.1.3 Command sequence grid

The command sequence grid is the workspace in the command sequence editor. Command sequences can be created with a graphical user interface here.

The diagram is divided into grids, with each grid offering room for one element.



TECHNICAL DETAILS

- Sheet size:
  - Default: 11 x 11 cells
  - Minimum: 5 x 5 cells

- Maximum: 500 x 1000 cells

- Cell size
  - Default: 155 x 111 pixels
- Outside edge: 100 pixel
- Grid: is displayed by default; can also be hidden
- Scroll bar: Is displayed if the document is larger than the frame.
- Scrolling via mouse wheel: up and down or if you press and hold `Shift` left and right.
- Zooming: `Ctrl + mouse wheel`
- Selecting elements: `left mouse click`
- Multiple selection: `Ctrl + mouse click`
- Move symbol: Click element and move it over the diagram while holding the left mouse button pressed. Content can be dropped to cells with green background. If a cell turns red when you move over it, you cannot drop the content.

10.1.4 Modes

The command sequence editor has several modes:

- Edit mode (on page 29)
  Command sequences are configured and edited in this mode
- Test mode (on page 58)
  Mode for testing and execution of the configured command sequences.
  The test mode has two modes:
  - Automatic mode
  - Semi-automatic mode

Command sequence - test mode

The command sequence editor executes command sequences in Runtime. You can start any number of command sequences. Test mode is for testing a command sequence but also to execute this. In addition in the test mode changes in the Editor can be applied directly via reloading the Runtime.

Exception: If a command sequence is currently being executed, the reloading of this command sequence is delayed. The reloading process is only carried once the command sequence has finished, been stopped or been canceled.

The following modes are available for execution in test mode:

- Automatic mode (on page 29)
  The configured command sequence is executed in automatic mode. The command sequence is only stopped in the event of pending user interaction.
- Semi-automatic mode (on page 29)
  There is a pause after each step in semi-automatic mode. A jump to the next step is only made after a corresponding click Configuration can thus be tested step by step.

> **Information**
>
> The fundamental command sequence process cannot be changed in test mode. You can only change values of the command tags.

Automatic mode

A configured command sequence runs in automatic mode. This mode is also used to visualize and control a configuration in Runtime.

If user interaction is necessary, the configured command processing screen is called up. The command sequence continues after an entry is made in the command processing.

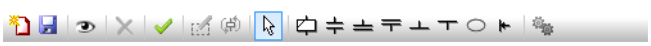
Semi-automatic mode

The configured command sequence is executed in automatic mode. The command sequence is only stopped in the event of pending user interaction.

Only after corresponding user interaction on the **`Continue command sequence at all execution positions`** button or **`Only continue command sequence at corresponding execution positions`** does the next step become active. This mode is thus suitable for stepping through a command sequence step by step.

10.1.5 Tool bar - command sequence editor (edit mode)

In edit mode of the command sequence editor, you can easily configure a command sequence in Runtime directly by means of drag&drop.

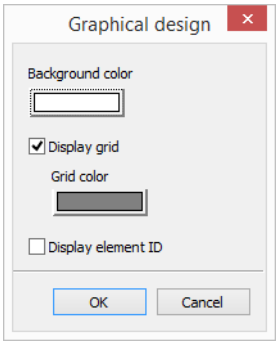


Parameters	Description
New command sequence	Creates new command sequence.
Save command sequence	Saves configured command sequence
Graphical design	Calls up dialog to select the graphical design (on page 31). The following can be selected: ▶ Background color ▶ Grid On/Off ▶ Grid color ▶ Display element ID
Delete	Deletes selected element Note: only active if an element was selected in the command sequence grid.
Check command sequence for errors	Checks configuration (on page 65) of a command sequence for logical correctness and consistency. The result is displayed in a dialog. ▶ `No errors during this command sequence.` ▶ `Checking the command sequence resulted in the following warnings/errors: {error details}`
Edit element	Calls up a dialog to change the element property for switchgear assignment (on page 41) and transition (on page 36). Note: not available for step.
Replace step	Opens the dialog (on page 35) to select a command processing action. As a result of this, already-configured steps can be assigned new actions. Note: only active if an element was selected in the command sequence grid.
Edit mode	Switches the mouse cursor from adding an element to edit mode. The switch back to the edit mode can also be achieved by pressing the `Esc` key.
Add step	Occupies the mouse pointer with a step. It can be added to any allowed, free location via click. Opens the dialog to select a command processing action.

Insert transition	Occupies the mouse pointer with a transition (on page 36). It can be added to any allowed, free location via click.
Insert Begin parallel branch	Occupies the mouse cursor with a begin parallel branch (on page 37). It can be added to any allowed, free location via click.
Insert End parallel branch	Occupies the mouse cursor with an end parallel branch (on page 37). It can be added to any allowed, free location via click.
Insert Begin branch	Occupies the mouse cursor with a begin branch (on page 38). It can be added to any allowed, free location via click.
Insert End branch	Occupies the mouse cursor with an end branch (on page 38). It can be added to any allowed, free location via click.
Insert switchgear allocation	Occupies the mouse pointer with a switchgear allocation (on page 41). It can be added to any allowed, free location via click.
Insert jump target	Occupies the mouse cursor with a jump target (on page 43). It can be added to any allowed, free location via click.
Switch command sequence to test mode	Switches command sequence to test mode (on page 58).

Graphical design

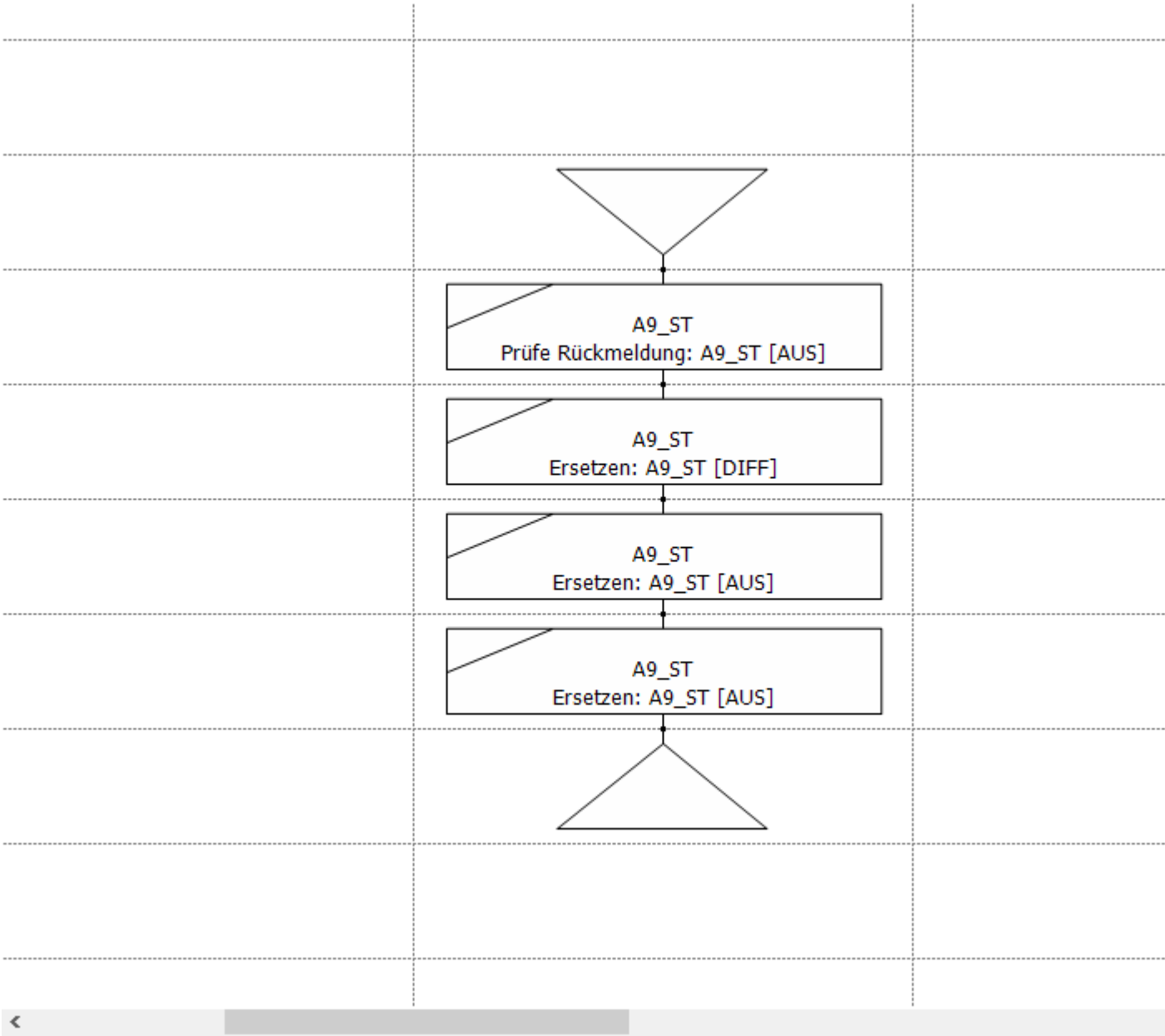
Clicking on the symbol for the **graphical design** in the tool bar opens the dialog for configuring the colors, grid settings and display of the element ID.



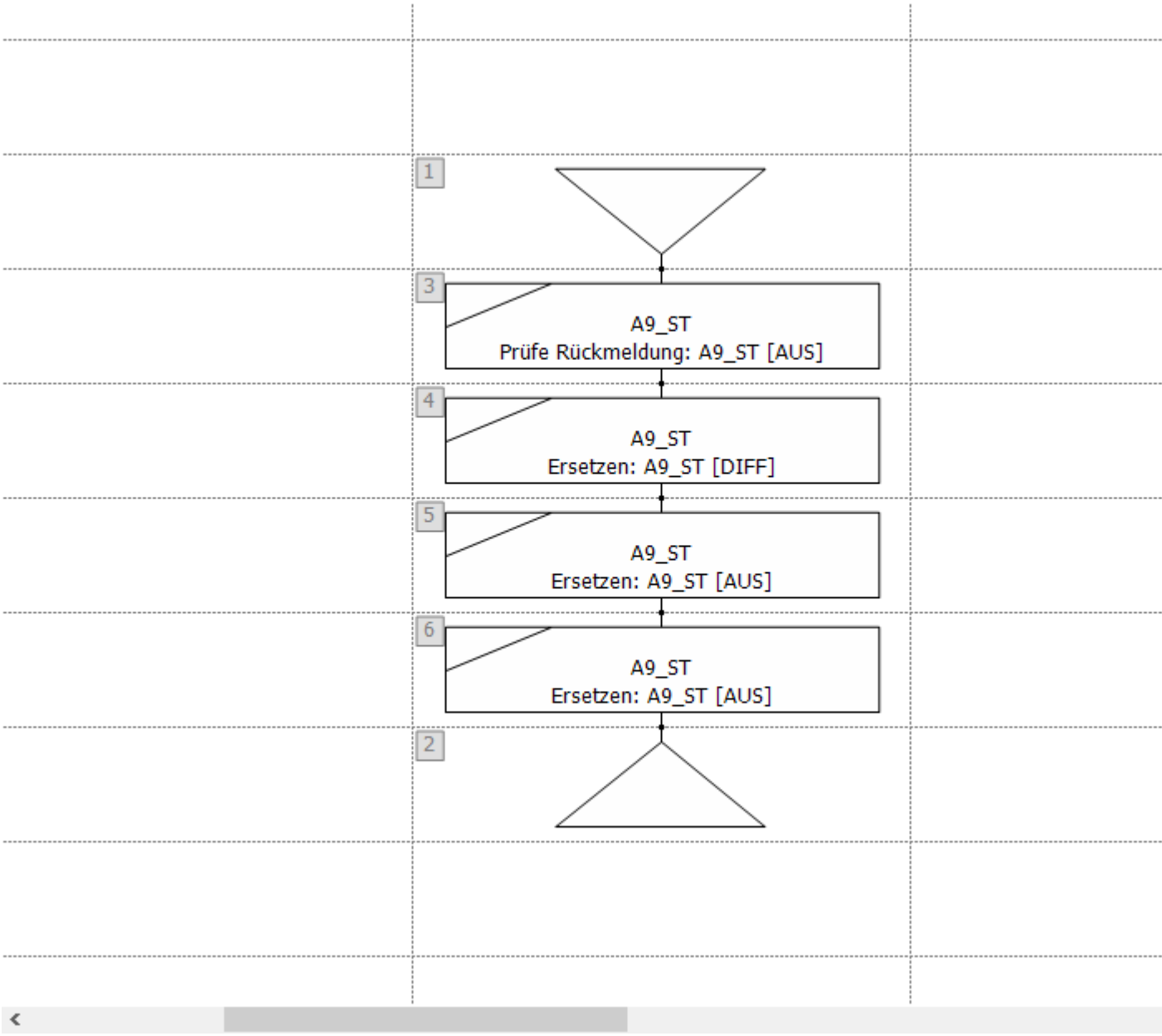
Parameters	Description
Background color	Defines the background color of the command sequence grid. Click on the color in order to open the palette for selecting a color.
Display grid	▸ `Active`: Display the grid ▸ `Inactive`: Grid is hidden.
Grid color	Defines the line color of the grid. Click on the color in order to open the palette for selecting a color.
Display element ID	Shows or hides the element ID. This setting is to be made for each command sequence. `Inactive`: No element ID is shown in the command sequence. Note: This setting is recommended for normal operation. `Active`: The ID of the elements is displayed in the command sequence. The exception is lines. The display is in the upper left-hand corner of the element. Hint: This setting is recommended for troubleshooting.
OK	Applies all settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

EXAMPLES

DISPLAY IDS OF THE ELEMENTS ACTIVE


A9_ST
Prüfe Rückmeldung: A9_ST [AUS]
A9_ST
Ersetzen: A9_ST [DIFF]
A9_ST
Ersetzen: A9_ST [AUS]
A9_ST
Ersetzen: A9_ST [AUS]

DISPLAY IDS OF THE ELEMENTS INACTIVE



Elements

The following elements can be used for the configuration of a command sequence in the command sequence editor:

- Step (on page 35)
- Transition (on page 36)

- Parallel branch (on page 37)
- Branch (on page 38)
- Switchgear allocation (on page 41)
- Jump target (on page 43)
- Lines

START AND END ELEMENT

Each command sequence must have a start and end element.

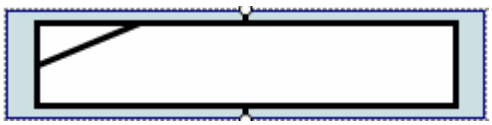
These two elements are automatically created when a command sequence is created and cannot be deleted from the project configuration. Even if you have configured the complete project configuration and deleted it, the start element and the end element are not affected by this deletion.

CONFIGURATION OF AN ELEMENT:

- Set the command sequence editor to edit mode.
- Select an element in the command sequence editor with a mouse click.
- Position this element in the command sequence editor by means of drag&drop.
  Note: You can position the element several times.
- Clicking on the **Esc** button deactivates the drag&drop functionality.

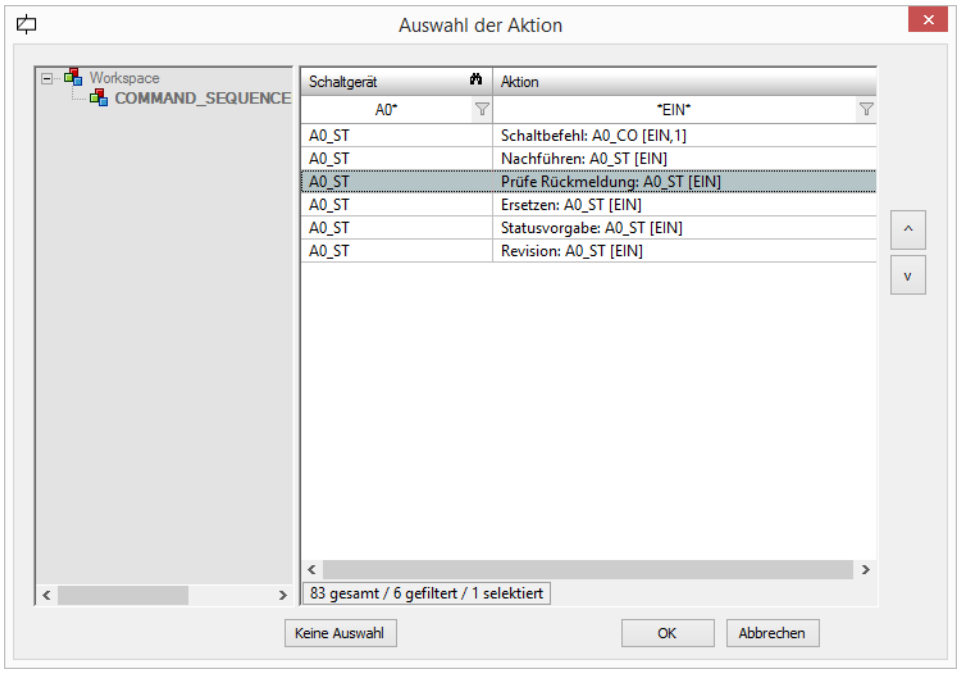
Step

A step is always linked to a configured action of command processing in the Command Sequencer module. The actions are configured in the zenon Editor in the Command Processing module.



After positioning of a step in the command sequence grid, the **Select the action** dialog opens automatically. In this dialog, you select an action from the list of all actions of all command groups configured in the zenon Editor.

ACTION SELECTION



Select an action and confirm the selection with OK. The command processing action is assigned to the step as a result.

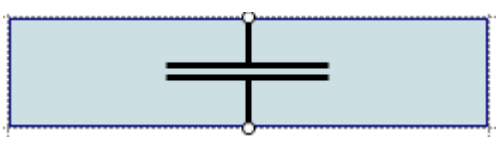
Note: The OK button is active if no action has been selected. Only the linking of an action is permitted for a step.

Information

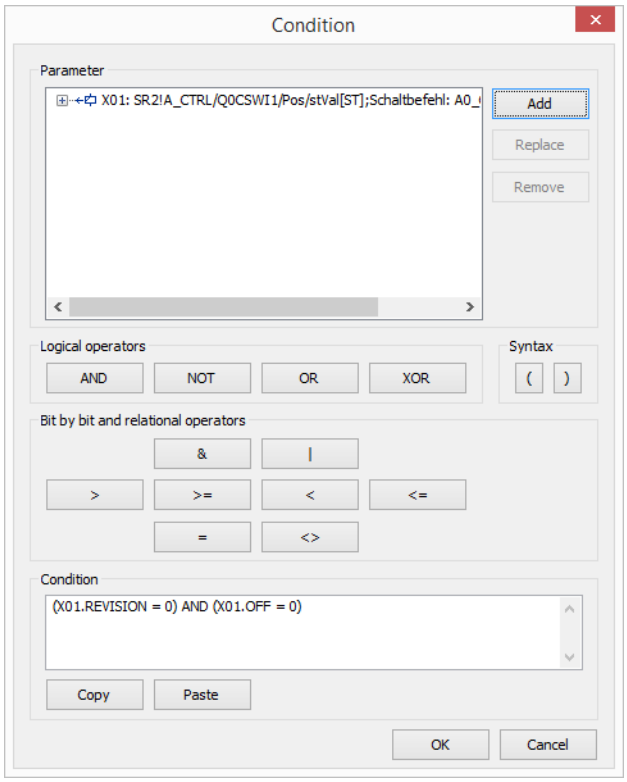
You can find further information on configuration of the command processing in the Energy Edition manual in the Command processing chapter.

Transitions - conditions

Transitions are used after steps in order to ensure a defined transition from one step to the next. Transitions display their internal status during the process and inform via a tool tip about status and process duration.



After positioning a transition in the command sequence grid, the **Condition** dialog opens automatically.



Select a response variable from the list of parameters. You get this list if you click on the **Add** button.

This list contains the response variables of all steps that have already been inserted into the command sequence grid.

Information

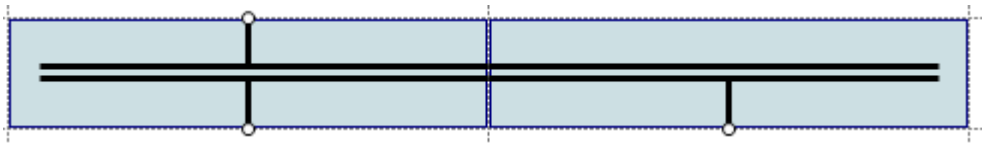
You can find further information on conditions in the Formula editor (on page 45) chapter in this manual.

Parallel branches

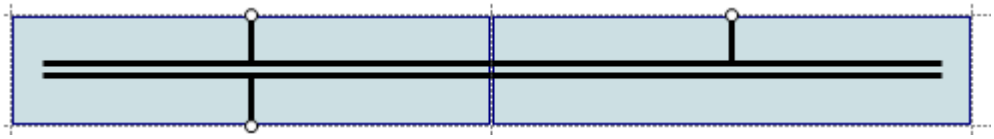
At the parallel branch an execution path parts into several execution paths which are executed in parallel during the process. For the activation of the different elements within a parallel branch you cannot define a certain order.

The project configuration always consists of a **Begin parallel branch** and an **End parallel branch**.

BEGIN PARALLEL BRANCH



END PARALLEL BRANCH



In the process the respective intermediate area of the **end parallel branch** is also colored. The color corresponds to the coloring of the command sequence.

A parallel branch is ended if the process has been completed in all execution paths. Completed means that either the following step is active or the following transition is inactive.

INSERT PARALLEL BRANCHES

To create a parallel branch:

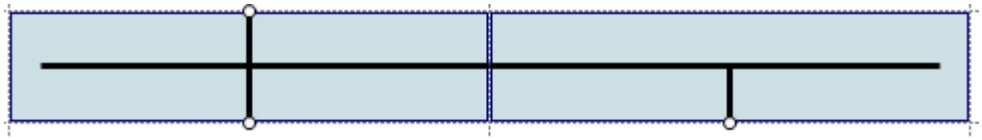
1. select the symbol **Insert begin parallel branch**
2. put the branch on the desired location
3. connect the input connection point with a output connection point of the preceding object
4. connect both output connection points with the desired following objects
5. close a parallel branch with object **Insert end parallel branch**

Branch

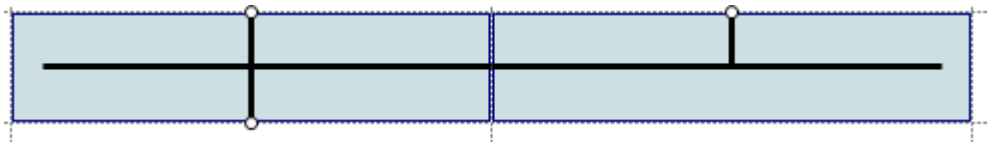
A branch offers the possibility to execute one of several possible paths. To do this, it is necessary that the first element at the start of a procedure path is a **transition**.

Note: A **Begin branch** can only be have a transition (on page 36) as a subsequent object.

BEGIN BRANCH



END BRANCH



PROCEDURE:

- The procedure path for which the first transition is `TRUE` is chosen.
- Then it is waited until all transitions have a value.
- If several transitions are `TRUE` at the same time, always the leftmost path for which the transition is `TRUE` is selected.

For begin and end the following is true: If there is a step in front of the element and a transition behind, the step remains active until the transition has been completed.

The objects are processed sequentially in a path. Each path processes its objects regardless of other paths.

Command sequences can select sequences and run in parallel branches (on page 37).

branches and parallel branches consist of:

- single/double horizontal lines
- Connection pieces (consisting of connection line and connection point)

CREATE A BRANCH

To create a branch:

1. Select the **`Insert begin branch`** symbol
2. Position the branch at the desired location.
3. Connect the input connection point to an output connection point of the preceding object.

4. Connect both output connection points with the desired following objects.
5. Close a branch with the **Insert end branch** object

Modify parallel branches and branches

MODIFY AND MOVE

Branches and parallel branches can be moved and changed in size.

MOVE

To move an object:

1. click on the object.
2. Hold down the mouse button.
3. Move the object to the desired position.

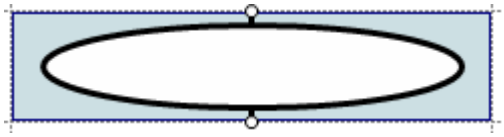
CHANGE SIZE

In this way object **Begin/End branch/parallel branch** can be extended and shortened. To change their size:

1. Move the mouse pointer over the object until it turns into a double arrow.
2. Hold down the left mouse button and move it in the desired direction:
   - Away from the object to extend it.
   - Into the object to shorten it.
   - The line to continue the command sequence remains unchanged each time.
3. at extending a new connection piece is added;
   - All fields in which lengthening is possible are colored green.
   - The process must be repeated to add several new connection pieces.
4. All corresponding connection pieces are deleted during shortening.

Switchgear allocation

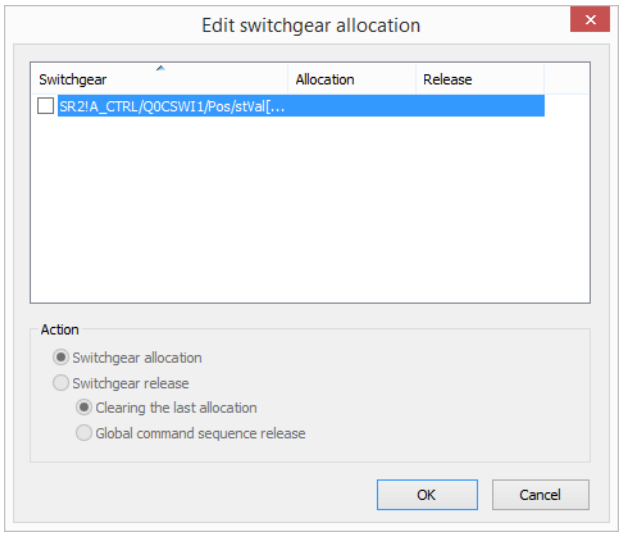
Each item of switchgear is represented by its response variable.



The **`switchgear allocation`** element of the module allocates one or more switching devices in Runtime:

- With this element, several (or all) response variables can be reserved (assigned) in advance. The NET_SEL status bit is set for this reservation.
- Unlocking is also carried out using a switchgear assignment element.

After the command sequence process, all NET_SEL status bits are automatically deleted again.



LIST OF SWITCHGEAR

Lists all available switchgear, its assignment and unlocking type. The list can be sorted - multiple selection is possible.

Parameters	Description
Switchgear	Switchgear according to configuration of the command processing in the zenon Editor.
Allocation	Yes, if **Switchgear allocation** is active. Empty if **Release** is active.
Release	Yes, if **Switchgear release** is active. In addition, the extent of the release is shown as text: ▶ `selective` ▶ `global` Empty if **allocation** is active.

ACTION

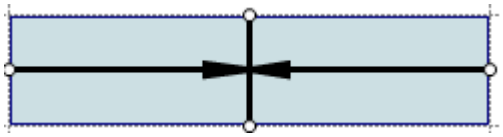
Parameters	Description
Switchgear allocation	If activated, the element is allocated to selected switchgear.
Switchgear release	If active, the element releases the selected switchgear.
Clearing the last allocation	If active, only the switchgear that was allocated in the last allocation is released. Only active if release for the switchgear is active.
Global command sequence release	If active, the switchgear that is allocated in the current allocation is released. Only active if release for the switchgear is active. Note: Allocated switchgear is automatically released again if the command sequence has been completed successfully or canceled.

CLOSE DIALOG

Parameters	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

Jump target

The jump target element allows a direct jump to a defined point of a branch.



Jump targets make it possible to

- To jump between procedure paths
- To jump out of branches
- engineer loops

Jump targets consist of tree inputs and one output. At this the output is always at the bottom and the inputs are located at the top and the sides. You can connect any input connection points. A path which ends in a jump target must have started with a **`Begin branch`**. Otherwise the end is not reached.

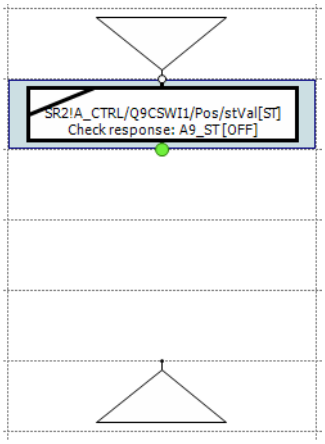
During the editing all connection points are visible. In the checking mode only the connection points which are connected are displayed.

> **Attention**
>
> Jump targets are not allowed for parallel branches.

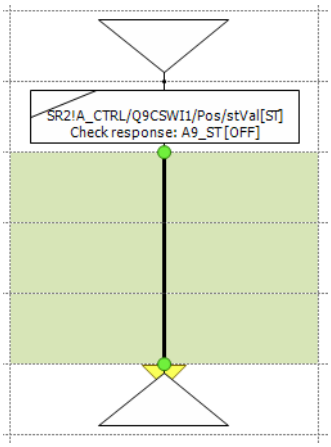
Lines

Lines connect elements via free connection points. To connect connection points with each other:

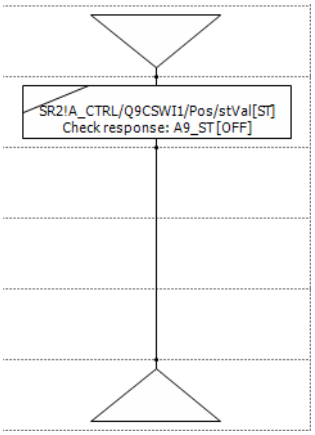
1. Activate a point with the help of the mouse:
   The connection point turns green. Red means that the connection point is already taken.



2. Drag a line to another connection point:
   A yellow arrow shows the direction of the line.. Green fields can be crossed. Red fields may not be crossed by the line.



3. As soon as the yellow tip of the line touches the next connection point, the line is created.



USING TREND CURVES

LINES:

- Are dragged with the mouse.
- Can be moved (press and hold `Ctrl` key)
  In doing so, all existing connections are separated and an attempt is made to reconnect the line if there are objects with connection points in the right direction at the target.
  If several lines are highlighted, the line that has a cell with mouse cursor in it is moved.
- Can be deleted by highlighting them and pressing the `Del` key.
- Are deleted when re-dragging them from the start to end.
- have a tool tip displaying its ID.

If a line reaches a connection point of an object, the connection point becomes active. If a connection is possible, it turns green otherwise red. Connections connecting two connections points of the same type - two inputs, two outputs, etc. - are not allowed. The line can be added in any case. Not allowed connections are displayed in red and trigger a corresponding error message at testing.

The connection points of the elements are always displayed in the edit mode even if the connection point in question is connected. In status "`Release`" no connection points are displayed.

Properties connection point:

- connected: highlighted red; connection is separated when the line is dragged and a new connection point can be chosen
- open: highlighted green; at dragging a new line is created

LINES CONNECT ELEMENTS

- Lines can be used as connections between all elements. It is allowed to add any number of lines after another.
- Lines must not be used to connect two equal connection points.
  For example: Both inputs of two steps must not be connected directly with a line. In the engineering this connection is allowed. It is however displayed in red (error) and in the validation (on page 65) an error message is displayed.

Formula editor

The formula editor is automatically opened if you need to enter or edit a formula. Above all:

Runtime:

- Editing transitions.

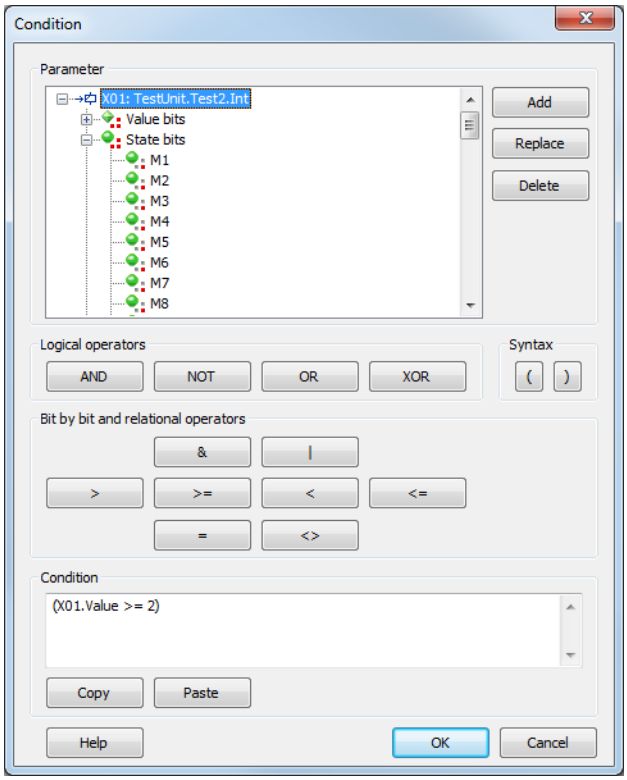
Note: If the step referenced in the formula is removed and a new step is added, the operands are reassigned in the case of operands. To do this, the same step must be reinserted. Parameters from a different phase are not automatically linked.

ENTER FORMULA

The following input is accepted:

- Constant as decimal number
- Hexadecimal number if it is preceded by an **x**
- Dot as decimal separator; the following is true:
  - Comma is automatically converted into a dot: 23,000 to 23.000

- Decimal places which are only zeros are removed: 23.000 to 23



Parameters	Description
TAG list	List of the tags which can be used for the formula. Each entry contains of: ▸ a basis node for the label ▸ a value ▸ a status ▸ the bits for value and status A symbol at the first node displays whether it is a command or return tag. The short indetifier at the beginning of the name is used for the formula.
Add	Opens the dialog for adding a parameter (on page 50). For this, the following applies: ▸ The following can be added: numeric and binary tags and tags for time duration. Values for duration are converted to seconds ▸ For conditions of a step, only the properties that were created for it can be used. ▸ Tags can be added multiple times.
Replace	Makes it possible to replace a tag. Clicking on the button opens the dialog to add a parameter (on page 50). ▸ Selection of a new parameter replaces the highlighted parameter. ▸ Clicking on the **no selection** button deletes the highlighted parameter from the list. The short identifier remains the same at replacing.
Remove	Removes the highlighted tag. For a tag to be deleted: ▸ the formula must be correct ▸ the selected tag must not be used in the formula
Logical operators	Via the buttons for operators, operators are added to the formula.
AND	logical 'AND'
OR	logical 'OR'
XOR	logical 'EXCLUSIVE OR'
NOT	Negation
Syntax	The operator buttons add the string shown on them to the formula.

(	Open parenthesis
)	Close parenthesis
Bit by bit and relational operators	
&	And
|	Or
>	greater than
>=	greater or equal
<	less than
<=	Less then or equal
=	equal
<>	less or greater
Condition	Configuration and display of the formula.
Copy	Copies the whole formula: ▸ All configured tags from the tag tree ▸ Formula from the field
Paste	Pastes a formula from the clipboard. At this all already configured elements are deleted and replaced by the copied formula. When copying formulas between steps, an attempt is made to resolve the operands via their names. For tags which are not found invalid entries are created in the operands list. Their point of use in the formula remain the same.
OK	Applies formula and closes the dialog. For this the formula must be correct.
Cancel	Discards all changes and closes the dialog.

Information

You can link up to 99 tags in a 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 tag value bit by bit. For binary tags only bit `0` is of importance, for SINT and USINT only the bits from `0 – 7`, etc.
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 52).
value and status	In the formulas, all values (value bits and status bits) are treated as binary values and can be logically linked with AND, OR, etc. The total value and overall status are an exception to this. In order to get a Boolean result this total value has to be ORed with a constant `bitwise` (on page 55). For this, we use the operator `&`. For the result 0 (false) of this logical ORing we get the binary value 0 (false), otherwise 1 (true). Example: see chapter Example bit by bit ORing (on page 55)

Info

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

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

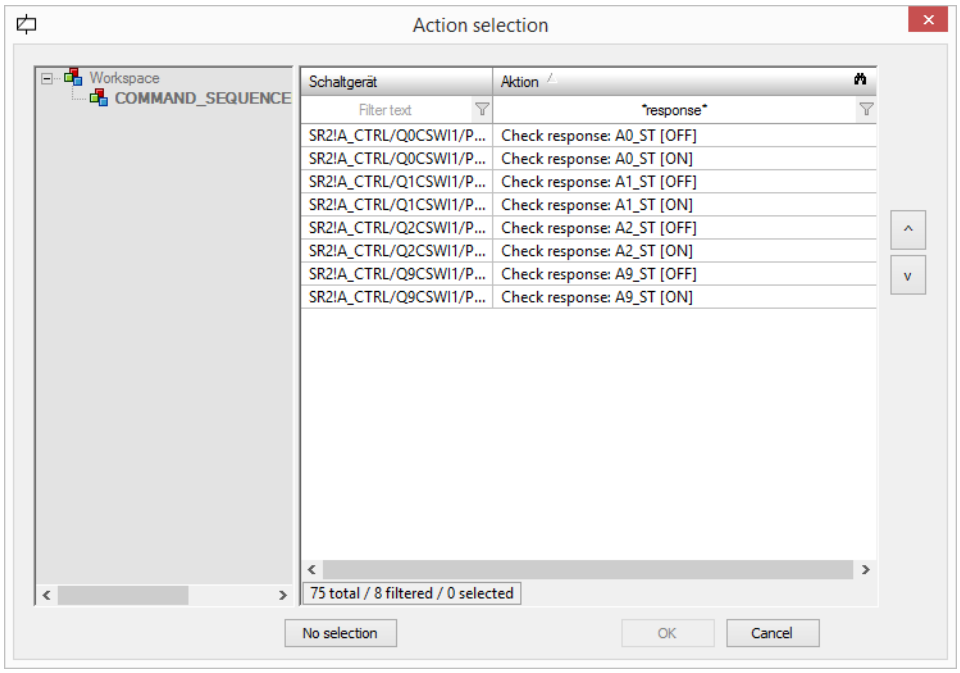
`X01.Value & X02.Value & X03.Value` -> does not work

But:

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

Adding parameters

Clicking on the **Add** button in the formula editor (on page 45) opens the dialog to select parameters that are to be used for a formula.



Parameters	Description
Project list	Display of the active project. Only parameters that have been created in the active project for the step to be configured can be selected.
Parameter List	List of the parameters available for the selected step. Multiple selection is possible. Apply by selecting and and clicking on the **OK** button or by double clicking on a parameter.
No selection	Deletes parameters already set. Only effective for replacement of parameters. If a parameter is highlighted in the formula editor and this dialog is opened by clicking on the **Replace** button, then clicking on the **No selection** button deletes the parameter from the list in the formula editor. The short identifier remains the same at replacing.
OK	Inserts selected parameters into the parameter list of the formula and closes the dialog.
Cancel	Discards selection and closes dialog.
Help	Opens online help.

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_INVAL	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.

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)
2	NOT
3	AND
4	XOR/OR

Info

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

Info

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

Bit formulas

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

Operator	Description
&	AND
|	OR

Example: ORing bitwise

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

USUAL FORMULA:

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

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

LOGICAL ORING:

```
X01.Status & 0xFF
```

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

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

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

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

Example:

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

```
X01.value = 202
```

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

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

Comparison operators

Comparison operators 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 OK (for example, 0x64 is in decimal figures 100).

Example

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

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

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

Examples for formulas

SIMPLE LOGICAL AND LINKING BETWEEN TWO BIT VALUES

Example

Formula: X01.03 AND X02.03

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

COMPARISON OF AN ANALOGUE VALUE OR STATUS OF A VARIABLE

Example

(X01.Value> X02.Value)

COMPARE ANALOG VALUES WITH EACH OTHER ON A LOGICAL BASIS

Example

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

COMPARE WITH VALUE BITS AND STATUS BITS

Example

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

COMPARE A VALUE WITH A DECIMAL OR HEXADECIMAL VALUE

Example

Formula: (X01.Value = 111)

Formula: (X01.Value = 0x6F)

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

Info

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

10.1.6 Tool bar - command sequence editor (test mode)

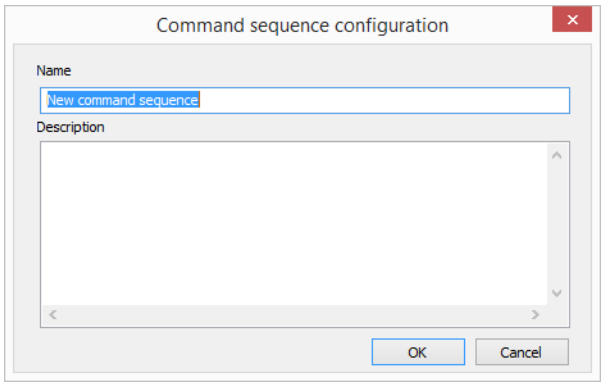
Parameters	Description
Start command sequence	Starts command sequence
Pause command sequence	Stops current command sequence and pauses it.
Continue command sequence	Continues paused command sequence. Only active if the command sequence was previously paused with **Pause command sequence**.
Abort command sequence	Aborts the command sequence that is running.
User interactions	Switches to the command processing screen. To do this, a step must be active in the command sequence. This step must be selected by clicking on the mouse. The screen that was linked in the command processing action is called up. If no screen is linked in the command processing action, the linked screen of the command processing group is used. Note: Only active is the action is being executed and a user interaction is expected, for example with a two-step action or an active interlocking.
Check command sequence for errors	Checks configured command sequence for logical consistency and possible errors. The result is displayed in a dialog. ▶ `No errors during this command sequence.` ▶ `Checking the command sequence resulted in the following warnings/errors: {error details}`
Edit element	Calls up a dialog with the element properties for switchgear allocation (on page 41) and transition (on page 36). No changes can be made in test mode. Switch the editor to edit mode in order to make changes. Note: not available for step.
Graphical design	Calls up dialog to select the graphical design (on page 31). The following can be selected: ▶ Background color ▶ Grid On/Off ▶ Grid color ▶ Display element ID
Switch to automatic mode	Switches execution of the switching step to `automatic`

	mode (on page 29).
Switch to semi-automatic mode	Switches execution of the switching step to semi-automatic mode (on page 29).
Continue command sequence at selected execution position only	Continues a command sequence from the selected element only. Note: only available in semi-automatic mode.
Continue command sequence at all execution positions	Continues a command sequence at all positions available - regardless of the respective position of the mouse pointer. Note: only available in semi-automatic mode.
Switch command sequence to edit mode.	Switches to edit mode - command sequences can be edited and repositioned.

10.2 Create command sequence

Command sequences are named and renamed in the **Command sequence configuration** dialog. A descriptive text can also be configured.

You create a new command sequence in the list of command sequences (on page 15).



Parameters	Description
Name	Name of the new command sequence. The name must not contain a question mark (**?**), a comma, an **@** or an asterisk (*****). Maximum length: 256 characters. Note: When duplicating command sequences, the existing name is supplemented with the prefix "**Copy of**". If the maximum length is exceeded by this, the name is shortened to the allowed length starting from the last character.
Description	(Optional) text for the description of a command sequence. You can change the description afterwards. To change the description, select the **Rename command sequence** symbol.

CLOSE DIALOG

Parameters	Description
OK	Applies settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

10.3 Tooltips

Tool tips in the command sequence editor visualize the respective status of a step and provide further information via the respective status (on page 62).

DISPLAY OF THE TOOL TIP:

To have a tool tip displayed, go to the respective step in the command sequence grid. The tool tip appears automatically when the mouse is positioned over the step.

STRUCTURE:

The tool tip consists of:

- Command processing command (command processing group action)
  - Internal ID

- Allocated switchgear

▶ General

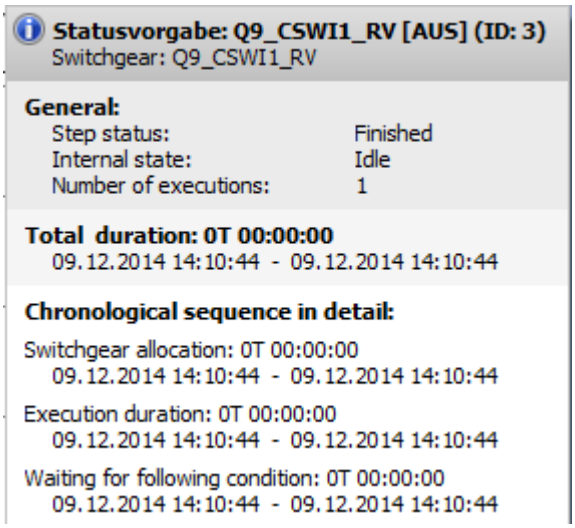
- Step status
- Internal state
- Number of executions

▶ Overall duration
(duration)

- Date and time stamp for start and end

▶ Chronological sequence in detail
(overall duration)

- Switchgear allocation
  Date and time stamp for start and end
- Duration of execution
  Date and time stamp for start and end
- Wait for subsequent condition
  Date and time stamp for start and end



10.4 Execution status

The following states are possible:

Status	Description
Idle	The command sequence is in idle state.
In execution	When starting a command sequence, it changes to running status.
Executed	As soon as the execution is finished, the command sequences switches to Finished status. In this status execution is not possible.
Pausing	The command sequence switches to paused status.
Paused	Within the phase, the process stops at: ▸ Waiting for Finished ▸ Waiting for **Allocation** ▸ Waiting for **Interlocking condition** ▸ Waiting for Phase finished ▸ Check for parallel execution
Aborting	Aborts the process and changes to Aborted.
Aborted	Command sequence process was aborted. If a command sequence cannot be restarted in the image on restarting, its status automatically changes to aborted.
Newly-occurred interlocking	Command sequence is stopped. Aborting the command sequence is now possible. This status occurs in the following scenario: ▸ An interlocking condition (one-step or two-step) waits for confirmation ▸ Whilst a confirmation is waited for, the active interlocking has changed again.

ACTIVE ELEMENT AND JUMP TARGETS

Status	Description
Continue	If an object is paused and an active element is located after it, continue has the same effect as Next step. This also includes jumps. For a step command, the command only affects the jump in the same branch.
Break	Has now effects for jump targets. Already defined targets remain.
Others	Always causes the deletion of the jumps. For a step command, only the jump in the area of the phase is deleted.

10.5 Symbols and Color

The states during the process of a command sequence are displayed using different symbols. Some symbols are also used for transitions and end parallel branch.

SYMBOLS AND WHAT THEY MEAN:

Symbol	Meaning
	Command sequence starts
	The connection is established.
	Wait for switchgear allocation. The switchgear of the step is already used in a different command sequence or is already assigned the `NET_SEL` status bit.
	▸ During the execution of a step and the waiting for **Reaction** `finished`. ▸ With transitions: Whilst running and waiting for transition condition. ▸ With end parallel branch: Waiting for all branches combined.
	Step has finished ▸ With transitions: Waiting for transition condition met. ▸ With end parallel branch: Waiting for all parallel branches finished and waiting for following condition.
	Values are written.
	User interaction required. calls up configured command processing screen.
	Command sequence is in semi-automatic mode and waits for the next step. To do this, click on the "**`Continue command sequence at all execution positions`**" button. It calls up the command processing screen (on page 71).
	Multiple executions. Occurs with an attempt to execute the exact same of steps at the same time.

If an error occurs during a step, the step is marked as faulty until it is restarted.

If a command sequence is paused, the current status is shown as a symbol.

STATUS

The execution status (on page 62) of **`steps`**, **`transitions`** and **`End parallel branch`** is signaled in color:

Status	Color
Idle:	White
In execution:	green
Finished:	blue
Pausing:	Two colors: ▸ orange ▸ Original color
Paused:	orange
Aborting:	Two colors: ▸ red ▸ Original color
Aborted:	red
Restarting:	Two colors: ▸ green ▸ Original color
Timeout:	red border
Newly-occurred interlocking	red border

ACTION ON STOP COMMAND

After a Stop command, the steps, transitions and end parallel branch immediately go to Stopped status, even if other elements are still waiting for a condition for stopping. Further subsequent commands such as Cancel are ignored. The Stopped status remains displayed.

10.6 Validate command sequence

Command sequences can be checked for errors during configuration.

- Validation is only possible in edit mode.
- Validation takes place automatically when switching from edit mode to test mode.

To validate a command sequence, click on the corresponding symbol in the tool bar of the command sequence editor in Runtime (green tick - **Check command sequence for errors**). The recipe thus is checked for functionality according to internal rules; the following in particular is checked:

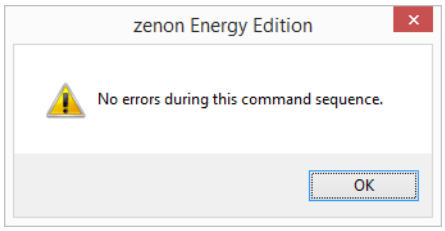
- Syntax (all lines connected, processable from begin to end, etc.)
- Variables
- Datatypes

The result of the check is displayed as a dialog in plain text. Found errors are also saved in the log file which can be analyzed with the Diagnosis Viewer.

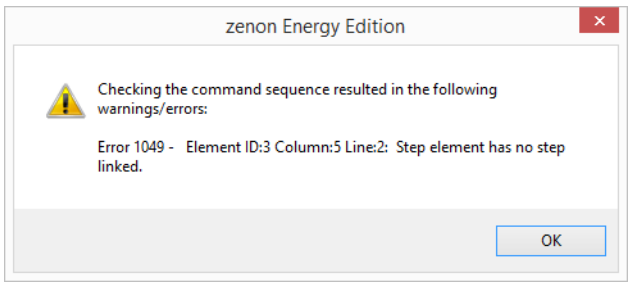
Rules that must be adhered to during configuration can be found in the Project configuration rules for recipes chapter.

> **Attention**
>
> *Command sequences that are not connected at the time of validation are ignored during validation. Their content and processes are not checked.*

DIALOG: VALIDATION OK



DIALOG: VALIDATION WITH ERRORS



If errors occur during checking, they are displayed in this dialog.

This error information includes:

- Error number
- Element ID
- Position in the command sequence grid
- Error text

10.7 Behavior of "Check response value" action type

The "Check response value" action type is used in the command sequences in particular to check the value of the response variable.

To be able to use the "Check response value" action type in the Command Sequencer module, this action type must be configured as an action in a command group in the Command Processing module in the zenon Editor.

> **Attention**
>
> This action type is not used in principle in the command processing. It was specially conceived for use in the Command Sequencer module.
>
> For command processing, it is possible to configure this query using action buttons and to receive responses in Runtime.

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

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

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

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

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

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

> **Information**
>
> *The **Check response value** action only serves to read the value of the response variable without executing an activity.*
>
> *The action is intended for use in the **Command Sequencer module**.*

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

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

10.8 Configuration rules for command sequences

The following important principles are applicable for configuration:

GENERAL

- For all elements all connection points must be connected.
  Exception: Jump targets. Only two of the three input connection points need to be linked there.
- The **`begin element`** is always present only once with a command sequence and marks the beginning of the process.
  Note: The begin element cannot be deleted.
- The **`end element`** is always present only once with a command sequence and marks the end of the process.
  Note: The end element cannot be deleted.
- Steps can be inserted anywhere. Several steps can also be placed in succession.
- There should be at least one active step in a command sequence.

TRANSITIONS

- Two transitions may not lie one after the other.

BRANCHES

- The first element after a **`Begin branch`** must be a transition.

- The individual branches which start at **Begin branch** must all end in an **End branch** never in an **End parallel branch**. Any element can be placed between begin and end of a branch even parallel branches as long as they are closed before the **End branch** element. An end branch can be replaced with jump targets at any point, including within a parallel branch.
- It is not necessary to have an **End branch** for each **Begin branch**. You can for example have two **Begin branch** elements ending in one **End branch** or the other way round.
- It is not necessary to have an **End branch** for a **Begin branch**. It can simply end in a line. If for example you have a **Begin branch** element with two paths and one of the paths ends in a jump target, it does not make sense to have an **End branch**.

PARALLEL BRANCHES

- Each parallel branch must contain at least one step.
- The first element after a **Begin parallel branch** must no be a transition.
- The individual branches that start at a **Begin parallel branch** must all end in one **End parallel branch**, but must never end in an **End branch**. You may use any elements between **Begin parallel branch** and **End parallel branch** even branches as long as they are closed before the **End parallel branch**.
- Not all branches which were started in a **Begin parallel branch** must end in an **End parallel branch**. It is enough when all branches converge over an **End parallel objects**. Equally branches from different **Begin parallel branch** objects may converge in a single **End parallel branch**.
- parallel branches allows embedding of additional parallel branches.
  In doing so: each embedded parallel branch must recombine with the superordinate parallel branch

LINES IN THE COMMAND SEQUENCE GRID

- Lines may be used as connections between any objects. It is allowed to add any number of lines after another.
- Lines must not be used to connect two equal connection points.
  For example: Both inputs of two steps must not be connected directly with a line. In the engineering this connection is allowed. It is however displayed in red (error) and in the validation (on page 65) an error message is displayed.

JUMP TARGETS

- Jump targets correspond to an **end branch**. They are intended to
  - jump between branches,
  - jump out of branches,
  - engineer loops

For this, the following applies: A path which ends in a jump target must have started with a **`Begin branch`**. Otherwise the end is not reached.

- Jump targets consist of tree inputs and one output. At least two inputs and the output must always be connected. At this it makes no difference which input connection point is connected.
- Jump targets can be switched consecutively if at least two input connection points are allocated.
- Jumps are prohibited:
  - between parallel branches
  - to jump out of a parallel branch
  - to jump in a parallel branch.

11. Command sequences in the zenon network

The Command Sequencer module is also available in the zenon network. In principle, command sequences are always executed on the Server in the process.

If the Server fails, the command processing screen is not visible on the Clients. Instead of the user interface, a blue area with an error message is shown.

Attention

The Command Sequencer module is not redundant!

If a computer works as a Client, all changes to the command sequence data is transferred to/from **Server 1**. If, when carrying out project configuration in the zenon Editor, a computer is configured as **Server 2**, it creates a local copy of the sequence data but is not capable of redundancy switching.

11.1 Particular aspects for the command processing screen

If user interaction has been configured in the command processing, the following is applicable:

- the command processing screen is automatically called up on the computer on which the command sequence was started.
- On all other computers, the command processing screen can be called up by clicking on the **User interaction** button.
- When executing a two-step step, a command processing screen is called up once the are no more active interlockings. The command processing screen is either that of an action or - if no screen has been configured there - the command processing screen of the command group.

12. Command sequences for zenon Web Server.

With a standard web client:

- The settings for grid and color can be changed
- No command sequences can be created or edited
- The size of the editing area cannot be changed
- In the tool bar, all symbols that are not permitted are deactivated;
  it is not possible to select the corresponding objects.

Web client PRO is not affected by these restrictions.