Manuel de zenon

Command Sequencer

v. 7.50

Table des matières

1. Bienvenue dans l'aide de COPA-DATA

AIDE GÉNÉRALE

Si vous ne trouvez pas certaines informations dans ce chapitre de l'aide, ou si vous souhaitez nous suggérer d'intégrer un complément d'information, veuillez nous contacter par e-mail : documentation@copadata.com (mailto:documentation@copadata.com).

ASSISTANCE PROJET

Si vous vous rendez compte que vous avez besoin de licences ou de modules supplémentaires, veuillezcontacter l'équipe commerciale par e-mail : support@copadata.com (mailto:support@copadata.com)

LICENCES ET MODULES

Si vous vous rendez compte que vous avez besoin de licences ou de modules supplémentaires, veuillezcontacter l'équipe commerciale par e-mail : E-mail 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 Command Processing - 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 Processing** module.

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

- Command
- Forced command
- Direct set value input
  - The set value configured in the command input can be transferred to the **Command Sequencer** module.
- Direct status input
- Replace
- Revision
- Direct correction
- Block
- Release
- Check response value
- Lock

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

3.2 Variables for command sequences

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 **Séquenceur de commandes** project property in the **Afficher les noms dans la grille de la séquence de commandes** 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 (à la page 68).

DISPLAY NAME IN THE COMMAND SEQUENCE GRID

The text display of a variable in Runtime is configured in the **Séquenceur de commandes** project property group in the **Afficher les noms dans la grille de la séquence de commandes** 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.

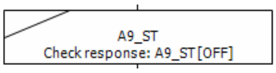
```
'<Variablename>' variable ignored for the command sequence data model because this
gives and invalid '<command sequence object name' object name!
Possible Reason: empty Identification or empty symbolic Address.
```

3.2.1 Examples of views: Display name in the command sequence grid

Examples of views for possible configurations of the variable names.

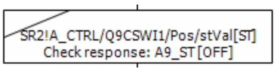
DISPLAY NAME IN THE COMMAND SEQUENCE GRID

VARIABLE NAME



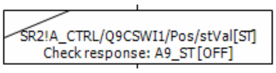
A9_ST
Check response: A9_ST[OFF]

VARIABLE IDENTIFICATION



SR2!A_CTRL/Q9CSWI1/Pos/stVal[ST]
Check response: A9_ST[OFF]

SYMBOLIC ADDRESS



SR2!A_CTRL/Q9CSWI1/Pos/stVal[ST]
Check response: A9_ST[OFF]

3.3 Display of the action name in Runtime:

The text display of a Command Processing action in the Runtime is configured in the project property group Comannd Sequencer in the **Nom d'étape dans la grille de la séquence de commandes** property.

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

Possible display names:

- Standard text
  Format: `Action name: RV[DIR]`
- Action name
  Name of the command processing action as configured in the **Nom de l'action** command processing property.

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 action created is unique. If this is not unique and there are two actions with the same name, an error message is given in the output window of the Editor:

```
The step '%s' for the switchgear '%s' will be ignored for the command sequence
model because the action name is not unique.
```

> **Attention**
>
> This error message means that the second step with the same name is not available in Runtime for the command sequences.

4. Terminology

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

Parameter	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 processing 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.
Teaching cursor	
Command sequence grid	
Project simulation	
Simulation mode:	
Simulation image	

5. Procedure

Configuration and 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 (à la page 34) 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 execution 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 processes of the module Command Processing in a graphic flow chart.

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

> **Informations concernant la licence**
>
> *Ces fonctions ou modules doivent disposer d'une licence pour l'Éditeur et le Runtime (simple, serveur, redondant et client).*

If both the Batch Control module and the Command Sequencer 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 **Paramètres du Runtime** project properties group.
- Select, for **Définition licence du module en runtime** of the **Module préféré** property, `command sequence`(default) or `Batch Control`.

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

PROJECT SIMULATION

The Project Simulation module, which requires a license, is included with the Command Sequencer module.

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 (à la page 7).
- Create a `command sequence` screen (à la page 14)
- In the `command sequence` screen, add the `command sequence editor` control element.

- Create a screen switch function (à la page 26) 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

CREATE A COMMAND SEQUENCE SCREEN

DÉVELOPPEMENT

Étapes de création du synoptique :

1. Créez un nouveau synoptique :

   Dans la barre d'outils ou le menu contextuel du nœud **Synoptiques**, sélectionnez la commande **Nouveau synoptique**.
   Un synoptique vide `Standard` est créé.

2. Modifiez les propriétés du synoptique :

   a) Nommez le synoptique dans la propriété **Nom**.

   b) Select `Command Sequencer` in the **Type de synoptique** property.

   c) Sélectionnez le cadre souhaité dans la propriété **Cadre**.

3. Configurez le contenu du synoptique :

   a) Sélectionnez l'option de menu **Éléments de contrôle** dans la barre de menus

   b) Sélectionnez `Insérer un modèle` dans la liste déroulante.
   La boîte de dialogue de sélection de mises en forme prédéfinies s'affiche à l'écran. Certains éléments de contrôle sont insérés dans le synoptique à des positions prédéfinies.

   c) Supprimez les éléments superflus du synoptique.

   d) Si nécessaire, sélectionnez des éléments supplémentaires dans la liste déroulante **Éléments**. Placez-les aux emplacements souhaités sur le synoptique.

4. Créez une fonction d'appel de synoptique.

7.2 zenon functions

The following functions are available for the **Command Sequencer** module:

- Execute command sequence or mode change (à la page 15):
  sends control commands to Command Sequencer execution

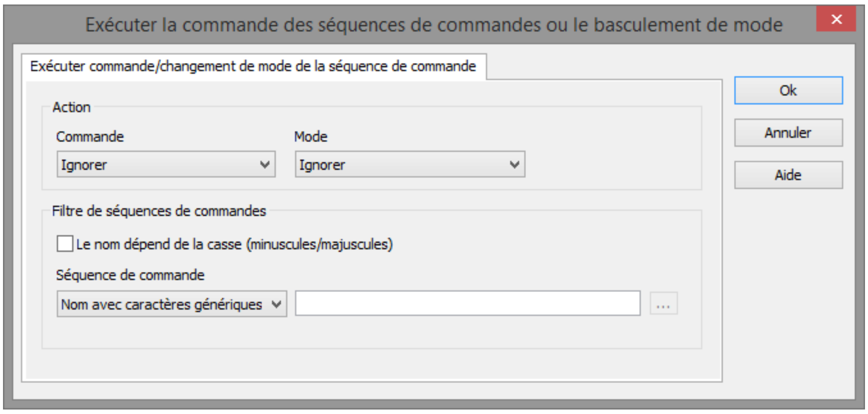
- Export comandd sequence (à la page 18):
  exports command sequences as an XML file. Content of the XML files can be filtered.
- Import command sequences (à la page 21)
  This function imports command sequences from a XML file.
- Teach command sequences (à la page 24)
  Starts or stops teaching mode.

7.2.1 Execute command sequence or mode change

You can send control commands to the Command Sequencer execution with this function.

To create the function:

1. In the zenon Editor, navigate to the **Functions** node
2. Select **New function**
3. Go to the **Command Sequencer** in the function selections
4. select **Exécuter la commande des séquences de commandes ou le basculement de mode**.
5. the dialog for configuring functions is opened



Exécuter la commande des séquences de commandes ou le basculement de mode

Exécuter commande/changement de mode de la séquence de commande

Action

Commande: Ignorer

Mode: Ignorer

Filtre de séquences de commandes

□ Le nom dépend de la casse (minuscules/majuscules)

Séquence de commande: Nom avec caractères génériques

ACTION

Parameter	Description
Action	Selection of the action to be executed: - Command - Mode
Command	Selection of the command to be executed from drop-down list: - `Ignore` Does not execute a command sequence command (`Start` or `Cancel`). Note: If automatic or semi-automatic is configured under mode, only one mode switch is executed. - `Start command sequence` Starts identified command sequence(s) - `Cancel command sequence` Cancels identified command sequence(s)
Mode	Selection of the mode in which the command sequence is to be executed: Selection of the mode from a drop-down list: - `Ignore` The command sequence(s) identified by the function is executed in the existing mode of the command sequence. - `Automatic` The command sequence(s) identified by the function is executed in `Automatic` mode. - `Semi-automatic` The command sequence(s) identified by the function is executed in `Semi-automatic` mode.

COMMAND SEQUENCE FILTER

Parameter	Description
Command sequence filter	Configuration of the command sequence filter. This determines the command sequences on which the function is to be applied.
Name is case sensitive	When filtering for command sequence commands, capital letters and small letters in the command sequence name are taken into account. Default: `Inactive`

Command sequence	Parameters for the selection of the command sequence. Select from drop-down list: ▶ `Name with wildcards:` A name with placeholder can be entered into the input field. Filtering is carried out according to this name. Wildcards are: `*`: any desired number of any characters `*` can be entered at the start, at the end or and the start and end in the input field. Note: The input field is only active with this option. ▸ `Name from variable:` The name of the command sequence is taken from a variable in Runtime. Click on button **...** opens the dialog for selecting variables. ▸ `ID from variable:` The ID of the command sequence is taken on by the ID from a variable. Click on the **...** button opens the dialog for selecting variables.

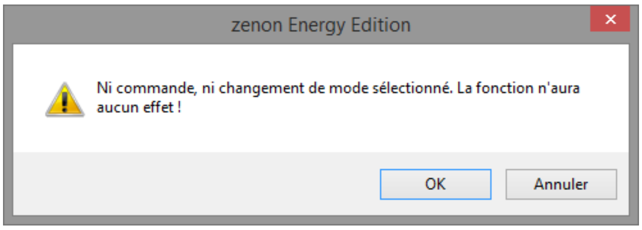
FERMER BOÎTE DE DIALOGUE

Paramètres	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

Informations

You can find detailed information for the configuration of the function in the configuration of the command sequence filter chapter.

ERROR DIALOG



zenon Energy Edition

Ni commande, ni changement de mode sélectionné. La fonction n'aura aucun effet !

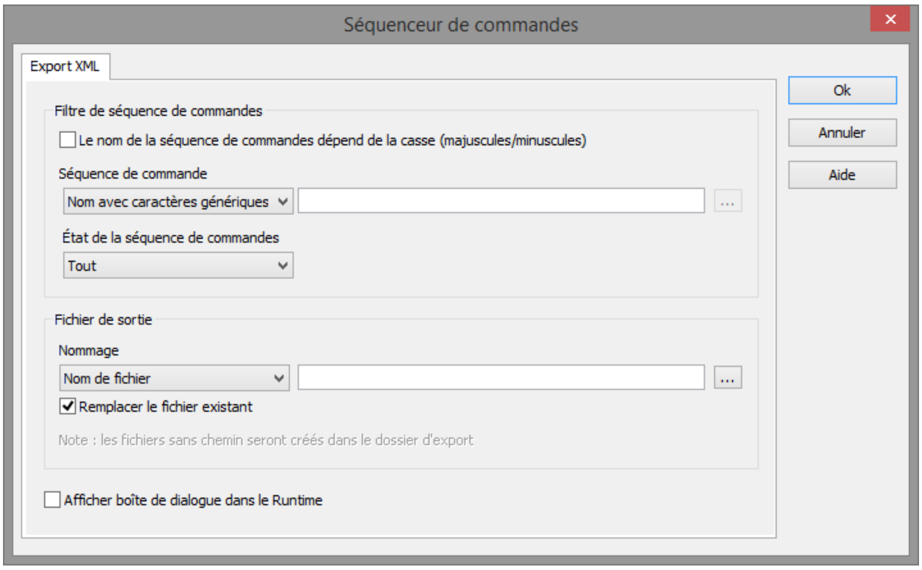
If, for **Command** and **Mode**, the `Ignore` property is configured for each, the function will not have an effect in Runtime. An error message is shown in this case.

7.2.2 Export command sequences

You export configured command sequences to an XML file using this function. The content of the export can be filtered.

To create the function:

1. In the zenon Editor, navigate to the **Functions** node.
2. Select **New function**.
3. Go to the **Command Sequencer** in the function selections.
4. Select **Exporter séquences de commandes**
5. The dialog for configuring functions is opened.



Filtre de séquence de commandes

□ Le nom de la séquence de commandes dépend de la casse (majuscules/minuscules)

Séquence de commande: Nom avec caractères génériques

État de la séquence de commandes: Tout

Fichier de sortie

Nommage: Nom de fichier

☑ Remplacer le fichier existant

Note : les fichiers sans chemin seront créés dans le dossier d'export

□ Afficher boîte de dialogue dans le Runtime

COMMAND SEQUENCE FILTER

Parameters	Description
Command sequence name is case sensitive	When selecting the command sequences to be exported, the capitalization of the command sequence name is taken into account.
Command sequence	- Name with wildcards Selection of the command sequences to be exported with the command sequence names. Entry of the search term in the input field. The following are wildcards: *: any desired number of any characters * can be entered at the start, at the end or and the start and end in the input field. Note: The input field is only active with this option. - Name from variable The name of the command sequences to be exported are taken from a variable in Runtime. Click on **...** to open the dialog to select a variable. - ID from variable The ID of the command sequences to be exported are taken by the ID from a variable. Click on **...** to open the dialog to select a variable.
Status command sequence	Selection of the status of the command sequences to be exported: - All All command sequences are exported. - Edit mode Only command sequences that are currently in edit mode are exported. - Execution mode Only command sequences that are currently in execution mode are exported.

OUTPUT FILE

Parameters	Description
Naming	Drop-down list to select how the output file is named: - File name: Name of the target file can be freely defined. Input of the file name in the input field

	- File name from variable: Name of target file is taken from a configured variable. Click on **...** to open the dialog to select a variable. - File name from command sequence name Name of the target file is named the same as the command sequence name to be exported - File name from command sequence ID File is named with the command sequence ID.
Overwrite existing file	Defines behavior if there is already an export file. If the checkbox is activated, the existing file is automatically overwritten (without another request for confirmation). Default: active
Show this dialog in the Runtime	Active : La boîte de dialogue de filtre est affichée dans le Runtime avant l'exécution de la fonction. Default: Inactive

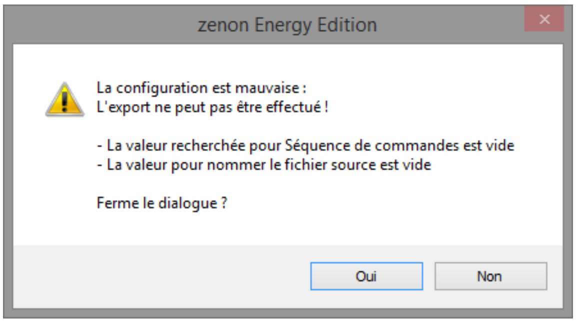
FERMER BOÎTE DE DIALOGUE

Paramètres	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l’aide en ligne.

ERROR DIALOG

A warning dialog appears in the event of an incorrect configuration of the export function:

- The search value for command sequence is empty:
  No command sequence for export could be found. Check the project configuration in the **command sequence filter** area.
- The value for the naming of the target file is empty:
  No command sequence for export could be found. Check the project configuration in the **output file** area.



zenon Energy Edition

La configuration est mauvaise :
L'export ne peut pas être effectué !

- La valeur recherchée pour Séquence de commandes est vide
- La valeur pour nommer le fichier source est vide

Ferme le dialogue ?

Note: The screenshot shows both possible causes of errors. Only one of the errors given can be displayed depending on the error.

Parameters	Description
Yes	The dialog of the function is closed. The dialog to configure the function is also closed.
No	The warning dialog is closed. The dialog to configure the function remains open for configuration of the function again.

Informations

You can get further information on the structure of the XML file in the Structure of the XML file for command sequences (à la page 115) chapter.

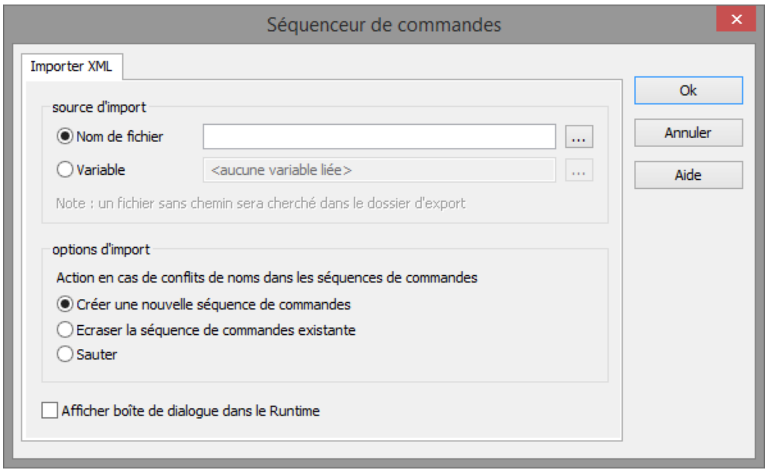
7.2.3 Import command sequences

You import command sequences from an XML file using the function.

To create the function:

1. In the zenon Editor, navigate to the **Functions** node.
2. Select **New function**.
3. Go to the **Command Sequencer** in the function selections.
4. Select **Importer séquences de commandes**.

5. The dialog for configuring functions is opened.



Séquenceur de commandes

Importer XML

source d'import

● Nom de fichier [ ] ...

○ Variable <aucune variable liée> ...

Note : un fichier sans chemin sera cherché dans le dossier d'export

options d'import

Action en cas de conflits de noms dans les séquences de commandes

● Créer une nouvelle séquence de commandes

○ Ecraser la séquence de commandes existante

○ Sauter

□ Afficher boîte de dialogue dans le Runtime

Informations

Existing command sequences are only overwritten on import if they are in the Command Sequences Editor.

More precise information on import can also be visualized using the two system driver variables **[command sequences] import event numerical** and -**[command sequences] import event string**.

IMPORT SOURCE

Selection of the import source for the import of a command sequence:

- File name
- From variable

Parameter	Description
File name	Selection of the XML import file. Clicking on **...** opens the dialog to select the file.
From variable	The name of the import file is taken from the selected variable. Click on **...** to open the dialog to select the variable.

IMPORT OPTIONS

Parameter	Description
Action for name conflicts in command sequences	Behavior in the event of naming conflicts: - `Create new command sequence` A new command sequence is created. Naming of the new command sequence: `[command sequence name from XML file]` + `[serial number].` - `Overwrite old command sequence` Existing command sequence is overwritten/replaced with content from XML import. - `Skip` No import is carried out for command sequences that already exist.
Show this dialog in the Runtime	`Active`: This dialog is opened in Runtime before the function is executed. Default: `Inactive`

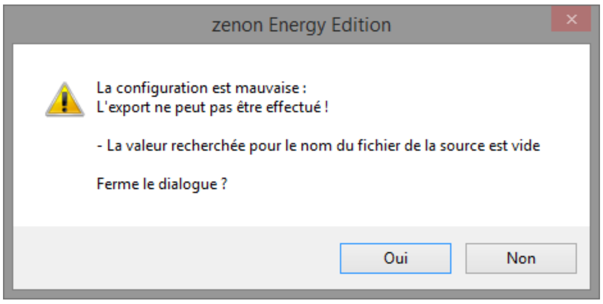
FERMER BOÎTE DE DIALOGUE

Paramètres	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

ERROR DIALOG

A warning dialog appears in the event of an incorrect configuration of the import function:

- `The search value for file name of the data source is empty:`
  No command sequence for import could be found. Check the project configuration in the **command sequence filter** area.


zenon Energy Edition

La configuration est mauvaise :
L'export ne peut pas être effectué !

- La valeur recherchée pour le nom du fichier de la source est vide

Ferme le dialogue ?

Parameter	Description
Yes	The dialog of the function is closed. The dialog to configure the function is also closed.
No	The warning dialog is closed. The dialog to configure the function remains open for configuration of the function again.

Informations

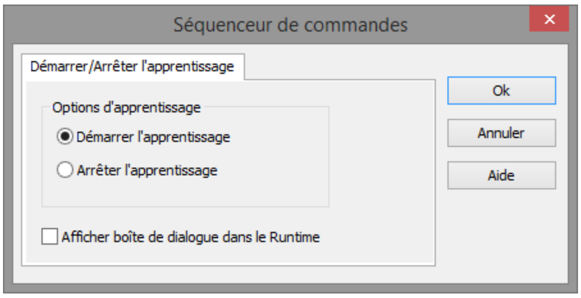
You can get further information on the structure of the XML file in the Structure of the XML file for command sequences (à la page 115) chapter.

7.2.4 Teach command sequences

Start or stop the teaching process in Runtime using this button.
This function is particularly suitable for starting teaching in Runtime in a process screen, without having to switch to the command sequence editor.

To create the function:

1. In the zenon Editor, navigate to the **Functions** node.
2. Select **New function**.
3. Go to the **Command Sequencer** in the function selections.
4. Select **Apprendre les séquences de commande**
5. The dialog for configuring functions is opened.



Séquenceur de commandes

Démarrer/Arrêter l'apprentissage

Options d'apprentissage

● Démarrer l'apprentissage

○ Arrêter l'apprentissage

□ Afficher boîte de dialogue dans le Runtime

Parameter	Description
Start teaching	Starts a new teaching process. In doing so, a new command sequence with a standard name is created. The teaching cursor is initially positioned in this command sequence in the command sequence grid under the start element. Note: Runtime must be in simulation mode for this! You can find further information in the Teaching (à la page 99) chapter.
Stop teaching	Ends teaching.
Show this dialog in the Runtime	Active: This dialog is opened in Runtime before the function is executed. `Start teaching` or `Stop teaching` can be selected in Runtime. Default: `Inactive`

FERMER BOÎTE DE DIALOGUE

Paramètres	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

7.3 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.

Informations

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

7.4 Command input actions in the Command Sequencer module

In this chapter, you are provided with additional information on the behavior of configured commands in the **Command Sequencer** module.

7.4.1 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.

Le type d'action **Vérifier la valeur de réponse** permet de vérifier l'état ON ou OFF des variables.

Lors de l'exécution de l'action **Vérifier la valeur de réponse**, la clé standard **Annuler** est déverrouillée dans la fenêtre de traitement de la commande.

Dans ce cas, selon la configuration de la **surveillance du Runtime**, un délai est appliqué jusqu'à ce que la valeur de la variable de réponse corresponde à la valeur de la direction de vérification définie dans la propriété d'action **Direction de commutation**. Si la valeur de vérification est ON, cette valeur est 1 ; si la valeur est OFF, la valeur est 0.

Si aucune surveillance du Runtime n'a été configurée (**Surveillance du Runtime**= Aucune), le délai d'attente défini (environ 24 heures) représente le délai d'attente maximum. Dans le cas contraire, l'action est terminée et le bit d'état TIMEOUT est défini pour la variable de réponse.

Après l'exécution de l'action dans la fenêtre de commande, si aucune autre action n'est disponible, cela est dû aux raisons suivantes :

- Le **délai d'attente** de la **surveillance du Runtime** n'est pas encore expiré.
- La variable de réponse n'a pas encore la valeur attendue (la modification de la valeur n'a pas encore été reçue).
- L'action n'a pas encore été annulée avec le bouton **Annuler**.

> **Informations**
>
> *L'action **Vérifier la valeur de réponse** sert uniquement à lire la valeur de la variable de réponse sans exécuter une activité.*
>
> *L'action est destinée à une utilisation dans le module **Command Sequencer**.*

Si la variable de réponse a déjà la valeur de la propriété **Direction de commutation**, l'exécution de l'action est reconnue comme terminée. Les autres boutons sur le synoptique de traitement de commandes sont ainsi immédiatement disponibles.

Remarque : Si la variable de réponse est définie sur OFF ou sur Revision , la valeur de réponse peut néanmoins être vérifiée.

7.4.2 Setpoint input command input action

If a **setpoint input** command is configured in the **Command input** module, a a set value of the command input can also be transferred in the command input as well as the **Command Sequencer** module.

WRITE SET VALUE WITH INPUT

Procedure for a setpoint entry with manual entry of value in the **Command Sequencer** module:

- When processing a command sequence, the command window is opened during the **Write set value with input** step: The **value of the command variables** element is available and filled with the current value. The **Execute** button is active. Action buttons are not active.
- The scroll bar is available and filled with the current value.
- The value is accepted after clicking on **Execute**.
- The execution of the switching sequences is continued with the new value.

WRITE SETPOINT VALUE WITH GIVEN VALUE

This value is configured in the Command Input module in the **Paramètres d'action** property with the **Spécifier valeur prescrite** property. If a value is to be proposed in Runtime, the **État de la réponse/direction de commutation** property in the zenon Editor must be configured with the `DIR` for the command. In this case, a value is prescribed in the Editor in the **Spécifier valeur prescrite** property.

TWO-STAGE EXECUTION

- The value is written after being confirmed with two-stage execution.
- The old value is still visible in the grayed-out input field whilst there is a wait for the confirmation or unlocking.

> **Informations**
>
> ***Write set value with entry of the value*** *is only possible from zenon 7.50.*
>
> **Informations**
>
> You can find further information in the Energy Editionmanual, in the Command input and Write set value action chapters.

7.4.3 Skip action for identical set value and actual value

If the **Ignorer l'action lorsque la valeur prescrite et la valeur réelle sont identiques** property in the zenon Editor has been activated, the values of the response variable are checked with the setpoint input during the process of the command sequence in Runtime.

The step is skipped if both values are identical. This is also shown accordingly in the tooltip (à la page 93) of the step (= `skip`).

8. System driver variables for the Command Sequencer module

The following system driver variables are available for this subject area:

Note: This group is only visible with a valid license for the **Command sequences** module.

Name	Data Type	Comment
[Command Sequencer] Number of pending user interactions	`DINT`	Number of command sequences with pending user interaction that are currently running. If the operation has been executed or the command sequence has been completed, the numeric value is reduced by 1. If several steps are waiting for an operation in a command sequence, the numeric variable is incremented for each step.
[Command Sequencer] Number of running command sequences	`DINT`	Number of command sequences currently running. The system variable is updated both at the start and end of a command sequence.
[Command Sequencer] Export result numeric	`DINT`	Result of the XML export: - `-1`: is being executed - `0`: Initialization value read successfully - from 1: Number of errors that occurred
[Command Sequencer] Export result string	`STRING`	Result of the XML export as a text. - `No errors occurred.` - `XML export error: The export file [save location]\[File Name] already exists and must not be overwritten.` Note: Only occurs if the **Overwrite existing file** property is not active in the export dialog and there is already a file with the same name in the export folder.
[Command Sequencer] Export result XML	`STRING`	Detailed content of the XML export. This variable visualizes the content of the XML export. The following are displayed: - `Name` - `Version` - `Type` - `ID` Note: If the content exceeds the maximum length of the system driver variable, the result is shortened.

Name	Data Type	Comment
[Command Sequencer] Import result numeric	`DINT`	Result of the XML import: - `-1`: is being executed - `0`: Initialization value read successfully - from 1: Number of errors that occurred
[Command Sequencer] Import result string	`STRING`	Result of the XML import as a text: - `The command sequence was not found. The command sequence therefore cannot be imported.` - `The command sequence could not be overwritten due to an incorrect status.` - `The command sequence cannot be imported as a new version. The versioning is not active.` - `The command sequence could not be imported. It does not match the selected type.` - `The command sequence could not be imported because the name is not permissible.`
[Command Sequencer] Import result XML	`STRING`	Detailed content of the XML import. This variable visualizes the content of the XML export. The following are displayed: - `Name` - `Version` - `Type` - `ID` Note: If the content exceeds the maximum length of the system driver variable, the result is shortened.
[Command Sequencer] Name of the active teached command sequence	`STRING`	Name of the command sequence that is currently being taught. The command sequence names are used once the teaching process has been started. If a teaching process has been completed, the value of this variable switches to empty.

Name	Data Type	Comment
[Command Sequencer] Names of the running command sequences	STRING	Names of the command sequences currently running: With several command sequences, the command sequence names are separated by a semicolon (;).
[Command Sequencer] Names of command sequences with pending user interaction	STRING	Names of command sequences running with user interactions pending. If several steps are waiting for an operation in a command sequence, the command sequence name is only entered once and is retained until all steps have been executed. With several command sequences, the command sequence names are separated by a semicolon (;).
[Command Sequencer] Teaching status	DINT	Status for the teaching process. Shows whether teaching is currently active or not active. ▶ 0 - Teaching is not active. ▶ 1 - Teaching cursor waits for positioning (this status is active until the teaching cursor in the command sequences editor has been placed) ▶ 2 - Teaching is active. This variable has the value 2 in the event of an ongoing teaching process. If a teaching process has been completed, the value of this variable switches to 0.

Informations

If a computer in redundancy operation upgrades to become the server, it sets the value of the system driver variables to `0` (numerical variables) or `empty string` (string variables).

Informations

You can find out further information in the system driver manual.

9. 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

To back up the Runtime files of configured command sequences:

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 or restarted again after copying.

10. Function authorizations

From zenon version 7.50, function authorizations are also supported by the **Command Sequencer** module.

They are configured in the **User administration** module. In doing so, a distinction is made between Editor authorizations and Runtime authorizations.
There are separate Runtime function authorizations available for the Command Sequencer module:

- Function authorizations for XML import and export
- Function authorizations to control the process of a command sequence

> **Informations**
>
> *You can find further information on the configuration of function authorizations in the User administration manual.*
>
> *You can find the special function authorizations for the Command Sequencer module in the Runtime function authorizations chapter.*

Attention

*Configurations of the **Command Sequencer** module always have priority over configured function authorizations for the Command Sequencer module during execution.*

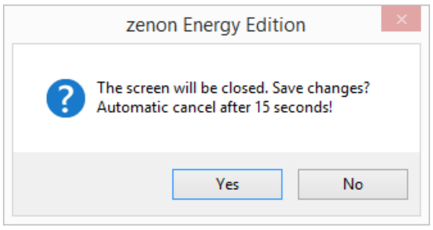
This ensures that there is no blockade during the course of a command sequence.

11. 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.



The screen will be closed. Save changes?
Automatic cancel after 15 seconds!

- 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.

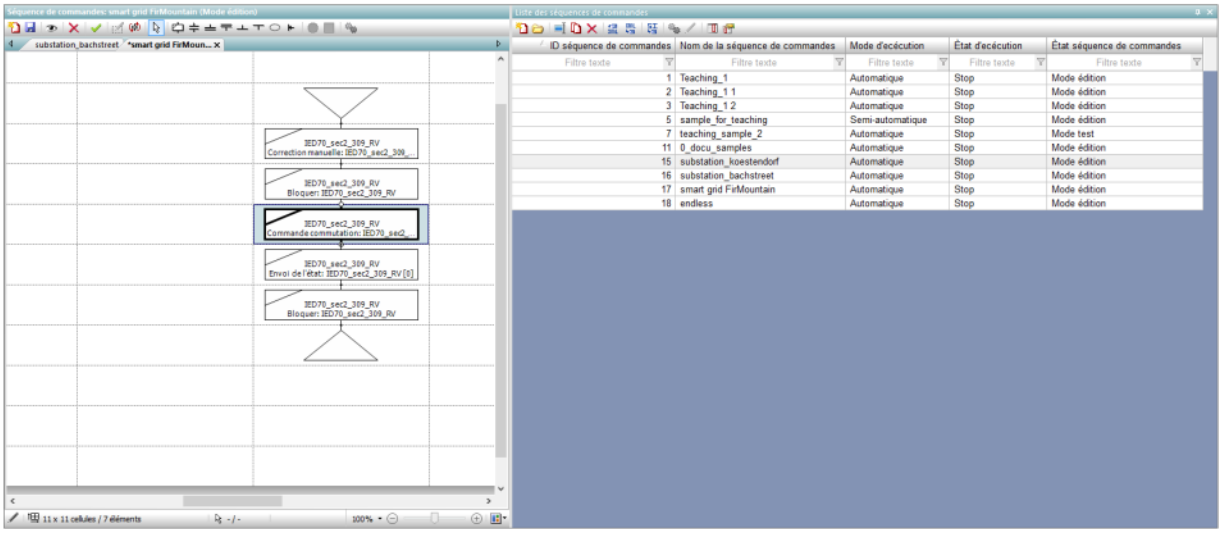
11.1 Command Sequencer Editor

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:

- Toolbars
  The design of the toolbars depends on the command sequence editor mode (à la page 58).

- Tabs
- Command sequence grid (à la page 53) project configuration area
- Dockable windows


Séquence de commandes: smart grid FirMountain (Mode édition)
substation_bachstreet
*smart grid FirMoun...
IED70_sec2_309_RV
Correction manuelle: IED70_sec2_309_...
IED70_sec2_309_RV
Bloquer: IED70_sec2_309_RV
IED70_sec2_309_RV
Commande commutation: IED70_sec2_...
IED70_sec2_309_RV
Envoi de l'état: IED70_sec2_309_RV [0]
IED70_sec2_309_RV
Bloquer: IED70_sec2_309_RV
11 x 11 cellules / 7 éléments
100%
Liste des séquences de commandes

ID séquence de commandes	Nom de la séquence de commandes	Mode d'ecécution	État d'ecécution	État séquence de commandes
Filtre texte	Filtre texte	Filtre texte	Filtre texte	Filtre texte
1	Teaching_1	Automatique	Stop	Mode édition
2	Teaching_1 1	Automatique	Stop	Mode édition
3	Teaching_1 2	Automatique	Stop	Mode édition
5	sample_for_teaching	Semi-automatique	Stop	Mode édition
7	teaching_sample_2	Automatique	Stop	Mode test
11	0_docu_samples	Automatique	Stop	Mode édition
15	substation_koestendorf	Automatique	Stop	Mode édition
16	substation_bachstreet	Automatique	Stop	Mode édition
17	smart grid FirMountain	Automatique	Stop	Mode édition
18	endless	Automatique	Stop	Mode édition

Parameter	Description
Header of the editor	Information about: - Current command sequence - Mode
command sequence editor menu bar	Menu bar (à la page 60) with symbols to configure a command sequence.
Tabs with opened command sequences	Select the open command sequences by clicking on the respective tab. In execution mode, the color of the tab corresponds to the color of the execution status (à la page 94) 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 grid • Number of configured elements ▶ Execution mode: • Symbol for execution 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 (à la page 38) Selected dockable windows are shown or hidden.

11.1.1 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

Parameter	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

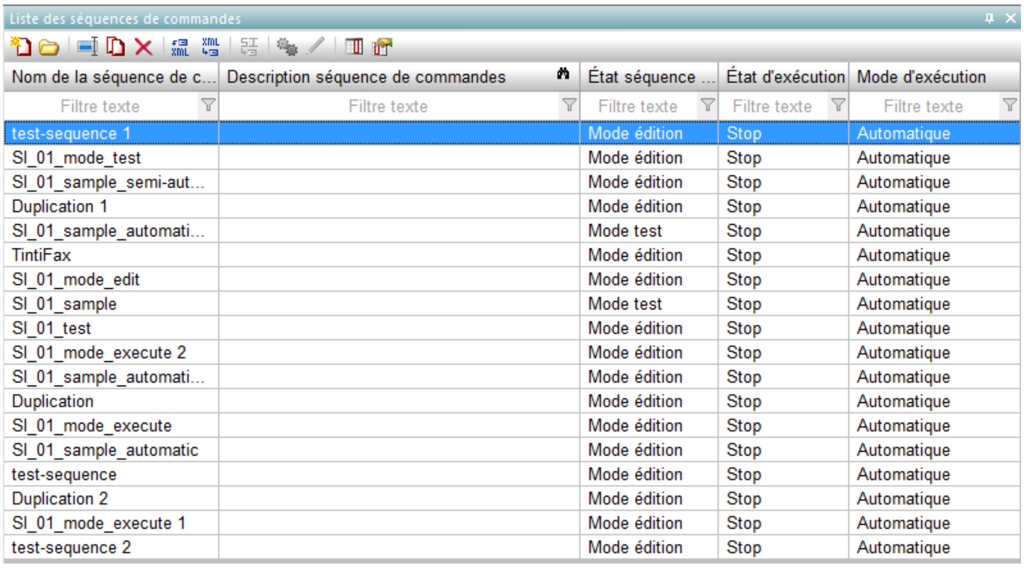
- Select the type of display:
  - Move display
  - Move display parallelly

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.

11.1.2 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.



Liste des séquences de commandes

Nom de la séquence de c...	Description séquence de commandes	État séquence ...	État d'exécution	Mode d'exécution
Filtre texte	Filtre texte	Filtre texte	Filtre texte	Filtre texte
test-sequence 1		Mode édition	Stop	Automatique
SI_01_mode_test		Mode édition	Stop	Automatique
SI_01_sample_semi-aut...		Mode édition	Stop	Automatique
Duplication 1		Mode édition	Stop	Automatique
SI_01_sample_automati...		Mode test	Stop	Automatique
TintiFax		Mode édition	Stop	Automatique
SI_01_mode_edit		Mode édition	Stop	Automatique
SI_01_sample		Mode test	Stop	Automatique
SI_01_test		Mode édition	Stop	Automatique
SI_01_mode_execute 2		Mode édition	Stop	Automatique
SI_01_sample_automati...		Mode édition	Stop	Automatique
Duplication		Mode édition	Stop	Automatique
SI_01_mode_execute		Mode édition	Stop	Automatique
SI_01_sample_automatic		Mode édition	Stop	Automatique
test-sequence		Mode édition	Stop	Automatique
Duplication 2		Mode édition	Stop	Automatique
SI_01_mode_execute 1		Mode édition	Stop	Automatique
test-sequence 2		Mode édition	Stop	Automatique

LIST OF COMMAND SEQUENCES - OVERVIEW:

The following is possible in the **list of command sequences**:

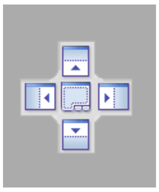
- New command sequences can be created.
- Command sequences that have already been configured in the command sequence editor are loaded for further editing.
- Command sequences are renamed.
- Command sequences are duplicated.
- Command sequences are deleted.
- Command sequences are switched to execution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

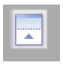
Selection and positioning

AIDES AU POSITIONNEMENT

Lors du déplacement de fenêtres depuis l'interface d'Editor, les aides au positionnement sont affichées. Celles-ci représentent les fenêtres ou leurs bordures.



Cet élément représente la zone d'une fenêtre dans Editor.



Cet élément représente la bordure d'une fenêtre dans Editor.

POSITIONNER LA FENÊTRE

Pour positionner un élément en tant qu'élément ancré :

1. A l'aide de la souris, déplacez l'élément vers la zone souhaitée...
2. L'aide au positionnement est affichée.
3. Ceci représente une fenêtre et ses zones :
   a) Centre : fenêtre entière

   b) Haut : moitié supérieure

   c) Bas : moitié inférieure

   d) Droite : moitié droite

   e) Gauche : moitié gauche

   ...Ou vers le bord de la fenêtre d'Editor.
4. Déplacez la souris vers l'aide au positionnement centrale ou vers une aide au positionnement située sur la bordure d'Editor, puis déplacez-la vers la zone souhaitée.
5. La zone d'Editor dans laquelle l'élément a été placé lorsque le bouton de la souris a été relâché est colorée en bleu.
6. Déplacez la souris dans l'aide au positionnement jusqu'à atteindre la zone souhaitée, affichée en bleu
7. Relâchez le bouton de la souris pour placer l'élément.

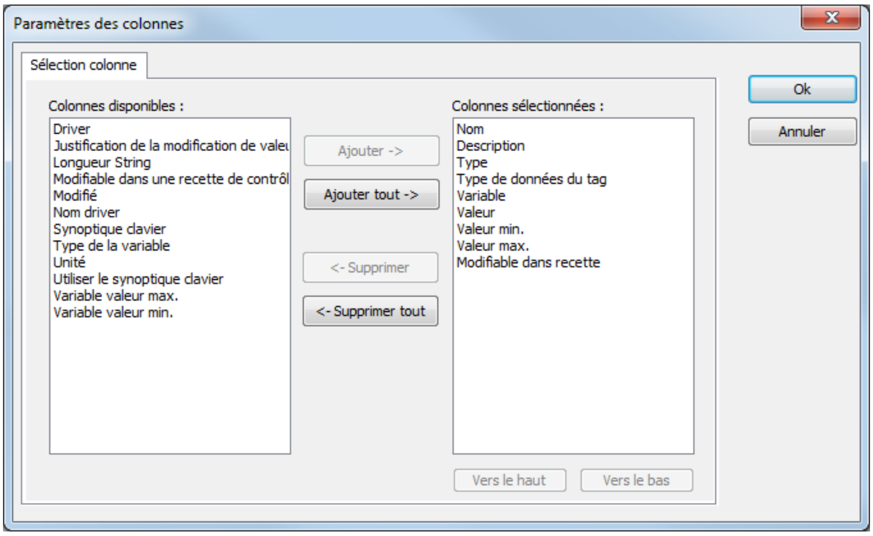
Si une fenêtre est placée sur une fenêtre existante, les deux fenêtres sont affichées au même emplacement, à l’aide d’onglets.

Toolbar - list of command sequences

Parameter	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 execution 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 dialog appears requesting confirmation of whether the selected command sequence(s) are really to be deleted for good.
Export selected XML	Exports selected command sequences as an XML file. Multiple selection is possible. Note: Not active if no command sequence is selected. You can find further information in the Export command sequence(s) as XML file (à la page 45) chapter.
Import XML	Imports command sequence(s) from an XML file. You can find further information about this in the Import command sequence(s) from XML file (à la page 47) chapter. Note: not active if the user does not have the corresponding function authorization. The **Command Sequences** function authorization:

	Import command sequences is applicable for both XML import and for the import of simulation images.
Import command sequence from simulation image	Imports command sequence from existing simulation images. You can find further information on this in the import command sequence(s) from simulation image (à la page 49) chapter. Note: not active if the project is running in simulation mode or if the user does not have the corresponding function authorization. The **Command Sequences** function authorization: **Import command sequences** is applicable for both XML import and for the import of simulation images.
Switch command sequence to execution mode	Switches selected command sequence(s) to execution mode (à la page 90). A validation of the configured command sequence is carried out automatically. Execution 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 execution mode back into edit mode (à la page 60) in order to make changes. Note: Not active if no command sequence is selected.
Column selection	Opens a dialog (à la page 42) to select columns that are to be displayed.
Column formats	Opens dialog (à la page 45) for the configuration of text and background colors for the display of columns in the list view.

Column selection



Colonnes disponibles :

Driver
Justification de la modification de valeu
Longueur String
Modifiable dans une recette de contrôl
Modifié
Nom driver
Synoptique clavier
Type de la variable
Unité
Utiliser le synoptique clavier
Variable valeur max.
Variable valeur min.

Colonnes sélectionnées :

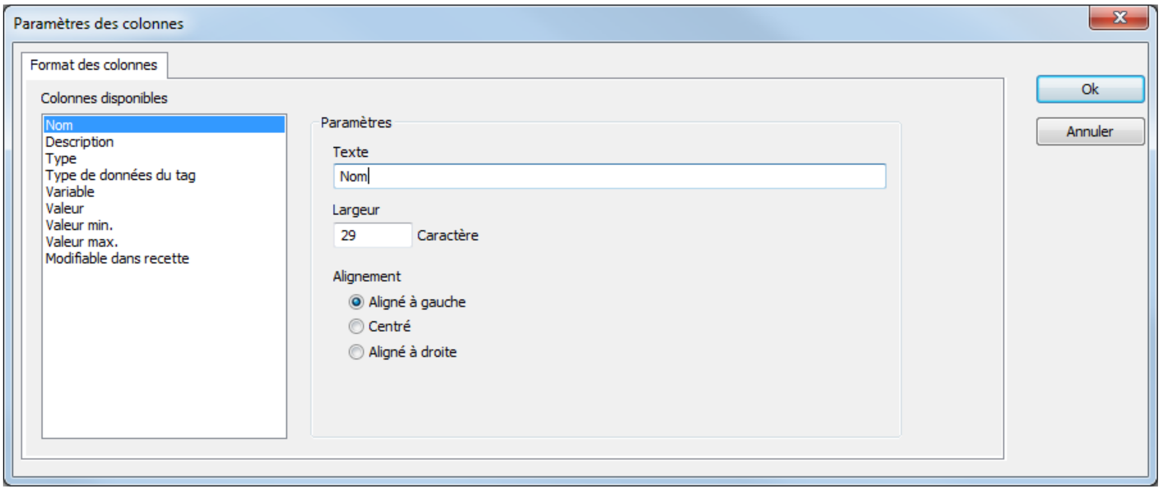
Nom
Description
Type
Type de données du tag
Variable
Valeur
Valeur min.
Valeur max.
Modifiable dans recette

Bouton	Fonction
Colonnes disponibles	Liste de colonnes pouvant être affichées dans la table.
Colonnes sélectionnées	Colonnes affichées dans la table.
Ajouter ->	Déplace la colonne sélectionnée des colonnes disponibles vers les éléments sélectionnés. Lorsque vous confirmez la boîte de dialogue en cliquant sur OK, ces colonnes sont affichées dans la vue de détail.
Ajouter toutes ->	Déplace toutes les colonnes disponibles vers les colonnes sélectionnées.
<- Supprimer	Supprime les colonnes marquées des éléments sélectionnés et les affiche dans la liste des colonnes disponibles. Lorsque vous confirmez la boîte de dialogue en cliquant sur OK, ces colonnes sont supprimées de la vue de détail.
<- Supprimer tout	Toutes les colonnes sont supprimées de la liste des colonnes sélectionnées.
Haut	Déplace l'entrée sélectionnée vers le haut. Cette fonction est uniquement disponible pour les entrées uniques ; les sélections multiples ne sont pas autorisées dans ce cas.
Bas	Déplace l'entrée sélectionnée vers le bas. Cette fonction est uniquement disponible pour les entrées uniques ; les sélections multiples ne sont pas autorisées dans ce cas.

FERMER BOÎTE DE DIALOGUE

Paramètres	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

Column Format



Paramètres des colonnes

Format des colonnes

Colonnes disponibles

- Nom
- Description
- Type
- Type de données du tag
- Variable
- Valeur
- Valeur min.
- Valeur max.
- Modifiable dans recette

Paramètres

Texte: Nom

Largeur: 29 Caractère

Alignement

● Aligné à gauche

○ Centré

○ Aligné à droite

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

Exporting the command sequence(s) as an XML file

You export selected command sequences as an XML file with the **Export selected XML** button. To do this, select the desired command sequences from the list of configured command sequences (multiple selection is possible). With multiple selection, only one XML file is saved for all selected command sequences.

NOTE DIALOG: EXPORT COMMAND SEQUENCE

Before export, a check is carried out to see if there are unsaved changes in a command sequence configuration.

> **Conseil**
>
> *Command sequences with unsaved changes are visualized in the tab of the command sequence editor with a* * *before the command sequence name.*

Parameter	Description
`[Name of the command sequence]`	Name of the command sequence that still has unsaved changes. It is always only the current command sequence name that is displayed. After clicking on the **Yes** or **No** button, the next command sequence with unsaved changes is displayed.
Yes	The current command sequence is saved before the save process. The save dialog is then opened.
No	Exports the displayed command sequence with the last-saved project configuration status. The save dialog is opened directly.
Yes all	All open command sequences with unsaved changes are saved before the save process. The save dialog is then opened. Note: not active if only one command sequence was selected for export.
No all	Exports all selected command sequences with the last-saved project configuration status. The save dialog is opened directly. Note: not active if only one command sequence was selected for export.

> **Informations**
>
> You can get further information on the structure of the XML file in the Structure of the XML file for command sequences (à la page 115) chapter.

Import command sequence(s) from an XML file

The **Import XML** button opens the dialog to import command sequences from an XML file in the list of command sequences.

Once the corresponding XML file has been selected, the import checks whether already-configured command sequences conflict with the command sequences of the XML file. In this case, the **Import options** dialog opens.

> **Informations**
>
> *Existing command sequences are only overwritten on import if they are in the Command Sequences Editor.*

Parameter	Description
[Name of the command sequence]	Name of the command sequence to be imported, which is already present in the **List of command sequences**. The first command sequence name is shown. After clicking on the **Skip** or **OK** button, the next command sequence is displayed with a naming conflict.
[Action in the event of naming conflicts]	Selection of the action for the import of a command sequence in the event of already-configured command sequences in the Runtime project configuration having the same name: - Create a new command sequence Creates a new command sequence in the Runtime configuration for the command sequence of the XML file. This imported command sequence is added when a new serial number is given: [command sequence name] [serial number] - Overwrite the existing command sequence Overwrites the existing command sequence in Runtime with the content of the XML file. Note: Grayed out if the file to be overwritten in the command sequence editor is in execution mode. Default: create a new command sequence
Carry out this action for all further conflicts (number of conflicts**)**	Carries out the import for all subsequent command sequences with naming conflicts with the action selected in [action in the event of naming conflicts]. The number in quotes states how many naming conflicts occur in the current import. There is no step-by-step display of the command sequences. Default: Inactive
Number of command sequences to be imported from the XML file:	Total number of command sequences for the import. Note: The number gives the total number of command sequences in the XML file, not just the number of conflicts.
Skip	Skips the import for the displayed command sequence. Switches to the next command sequence with conflicts in the event of several naming conflicts. Note: If the **Execute this action for all further conflicts** property has been activated, no command sequences with naming conflicts are exported.

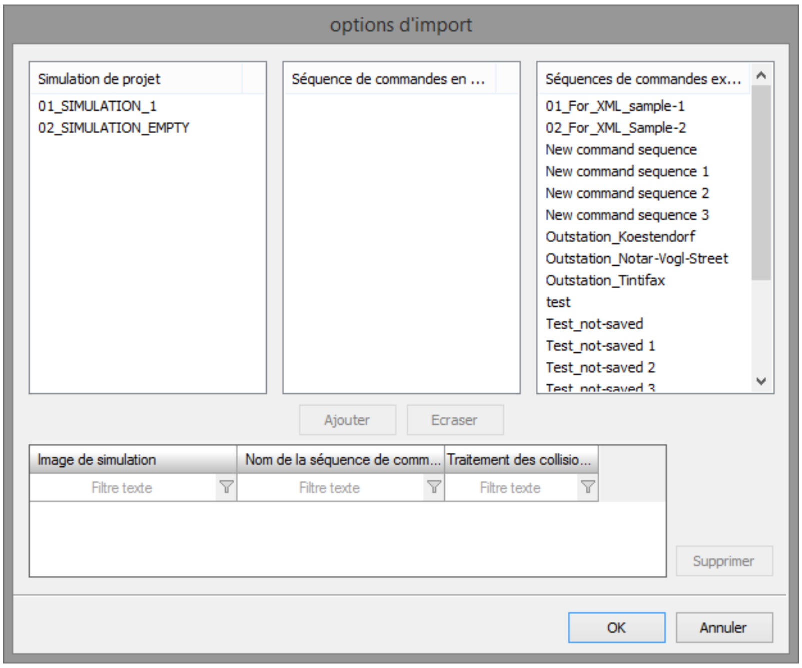
Parameter	Description
OK	Carries out the import for the displayed command sequence with the action selected with `[action in the event of naming conflicts]`. Switches to the next command sequence with conflicts in the event of several naming conflicts. Note: If the **Execute this action for all further conflicts** property has been activated, no command sequences with naming conflicts are exported.
Cancel	Cancels the import for all subsequent command sequences and closes the dialog.

Informations

You can get further information on the structure of the XML file in the Structure of the XML file for command sequences (à la page 115) chapter.

Import command sequence(s) from simulation image

In this dialog, you arrange configured command sequences from one or more simulation images for import into zenon Runtime.



01_SIMULATION_1
02_SIMULATION_EMPTY

01_For_XML_sample-1
02_For_XML_Sample-2
New command sequence
New command sequence 1
New command sequence 2
New command sequence 3
Outstation_Koestendorf
Outstation_Notar-Vogl-Street
Outstation_Tintifax
test
Test_not-saved
Test_not-saved 1
Test_not-saved 2
Test_not-saved 3

Parameters	Description
Project simulation	List of all existing project simulation images. Only one simulation image can be selected at a time.
Command sequences in project simulation	List of all configured command sequences from the selected project simulation. Multiple selection is possible.
Existing command sequences	Pre-existing command sequences in the zenon project that is currently running.
Add	Adds selected command sequences from the **Command sequences in project simulation** list to the `List of command sequences to be applied`. Only active if at least one command sequence from a simulation image has been selected.
Overwrite	Adds a command sequence from the **Command sequences in project simulation** list to the `List of command sequences to be applied`. This command sequence overwrites the existing command sequence in the command sequences editor in Runtime! This button is only active if just one command sequence has been selected and the selected command sequence is already in the **existing command sequences** list. The command sequence is thus already present in the command sequences editor and is overwritten. With multiple selection of command sequences from a project simulation, the button is then grayed out if the selection contains a command sequence that is not yet present in the list of **existing command sequences**. A new command sequence is created when transferring to Runtime. This new project configuration is given a serial number in the naming.
`List of the command sequences to be transferred.`	Assigned command sequences that are applied in the current Runtime environment. ▶ `Simulation image:` Name of the simulation image from which the command sequence comes. ▶ `Command sequence name:` Name of the command sequence as saved in the simulation image. ▶ `Conflict handling`: The type of conflict handling depends on how the respective command sequence is

	transferred to the list. Depending on the button used (**Add** or **Overwrite**), the conflict handling is prescribed and cannot be changed. Note: List can be sorted and filtered.
Remove	Removes highlighted command sequence from the List of command sequences to be applied. Multiple selection is possible. A new assignment from **Command sequences in project simulation** is possible.
OK	Closes the dialog and applies project configurations from the list of command sequences to be applied in the command sequence editor.
Cancel	Annule toutes les modifications et ferme la boîte de dialogue.

If the **Command Sequencer** module is operated in the zenon network, the following rules are applicable:

- If the dialog is called up, the **project simulation** list is filled with the simulation images from the server.
  If a server is lost, the dialog with an empty **project simulation** list is called up.
- The **command sequences in project simulation** list is also filled with data from the server on the client. The list is empty if the server is lost.

Toolbar - list of command sequences

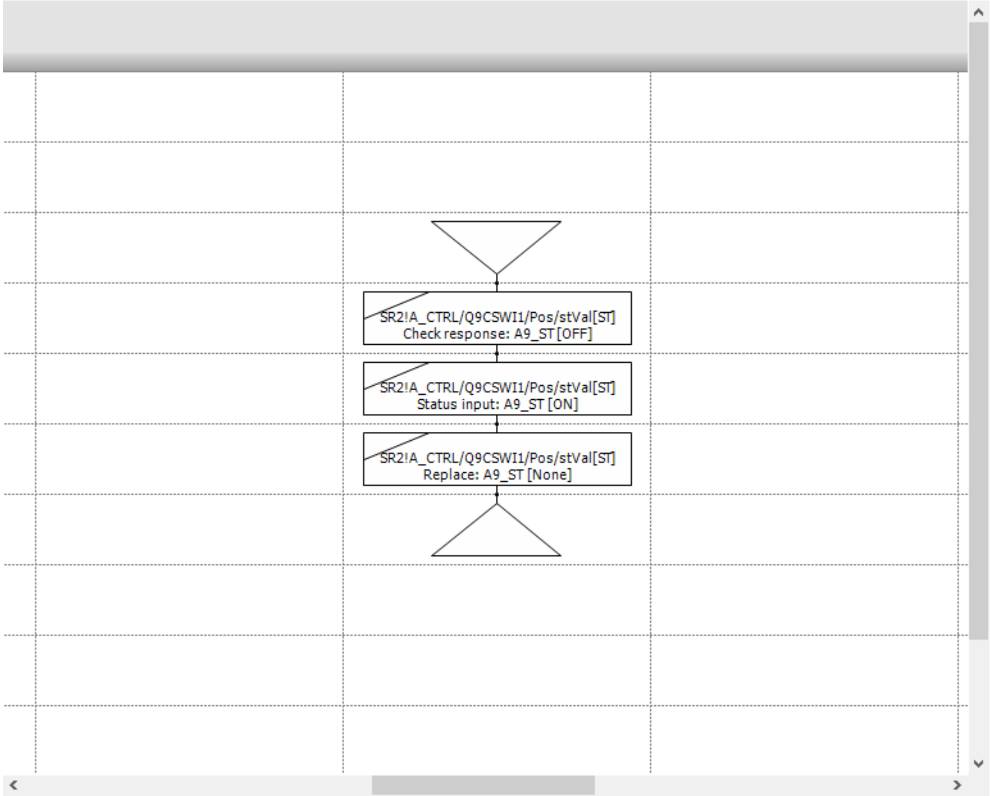
Parameters	Description
New command sequence	Creates an empty command sequence and opens the dialog to create a new command sequence (à la page 92).
Open in command sequence editor	Opens the selected command sequence in the command sequence editor (à la page 60). Note: Not active if no command sequence 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 execution 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.
Export selected XML...	Exports selected command sequences as an XML file. Note: Not active if no command sequence is selected.
XML import...	Imports command sequence(s) from an XML file.
Import from project simulation...	Imports command sequence from existing simulation images. You can find further information on this in the import command sequence(s) from simulation image (à la page 49) chapter. Note: not active if the project is currently running in simulation mode or if the user does not have the corresponding function authorization.

Switch to execution mode	Switches selected command sequence(s) to execution mode (à la page 90). A validation of the configured command sequence is carried out automatically. Execution 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 execution mode back into edit mode in order to make changes. Note: Not active if no command sequence is selected.

11.1.3 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.


SR2!A_CTRL/Q9CSWI1/Pos/stVal[ST]
Check response: A9_ST [OFF]
SR2!A_CTRL/Q9CSWI1/Pos/stVal[ST]
Status input: A9_ST [ON]
SR2!A_CTRL/Q9CSWI1/Pos/stVal[ST]
Replace: A9_ST [None]

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 with a 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.

AMEND SIZE OF THE GRID

The size of the grid can be amended. To amend the size of the grid:

- Move the mouse pointer to the upper, lower or right-hand edge of the grid.
  The mouse pointer changes at the edges: <=>
- Hold down the right mouse button and move the edges:
  The dimensions of the grid are reduced or enlarged.

Please keep in mind:

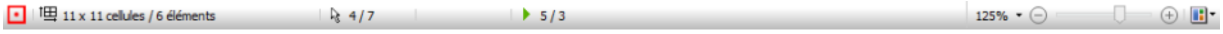
- The minimum size of the grid is 5 x 5 cells.
- A reduction can only be carried out to a maximum of the column or line in which an element has been placed.
  This guarantees that no existing project configuration is lost.
- The size of the grid is automatically enlarged during the teaching process if necessary.
- The number of new cells in the footer is visualized in the footer during the enlargement process:

Command sequence grid footer

The footer of the command sequence grid visualizes information for the configuration of the selected command sequence. The elements available are different depending on the mode.

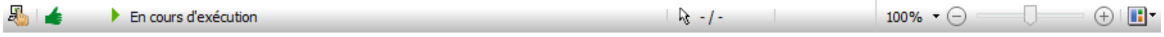
EDITING MODE VIEW

Footer of the command sequence grid in edit mode and if teaching mode has been activated:



EXECUTION MODE VIEW

Footer of the command sequence grid in execution mode:



Parameters	Description
Command sequence mode	Display of the active mode: ▶ Edit mode (symbol: pencil) Only available in edit mode. ▶ Teaching mode (symbol: red square) Only available in simulation mode. ▶ Execution mode: • Automatic mode (symbol: computer terminal) • Semi-automatic mode (symbol: computer terminal with hand symbol) You can find a graphic overview with examples of the icons in the Symbols of the footers (à la page 58) chapter.
Command sequence status of project configuration	Status of command sequence: ▶ Ready to start (symbol: green hand with thumb up) ▶ Warning (symbol: red hexagonal warning symbol with exclamation mark)
Process status	Graphic visualization for error-free and executable command sequence. ▶ Completed (symbol: green tick) ▶ Canceled (symbol: red X) ▶ In progress (symbol: green triangle) ▶ Idling (symbol: red minus sign) ▶ Paused (symbol: two red Is) Note: Only visible in execution mode. You can find a graphic overview with examples of the icons in the Symbols of the footers (à la page 58) chapter.

Size of the element and number of elements	Additional information on the command sequence grid: - Total size of the working area of the command sequence grid. `[number of horizontal cells] x [number of vertical cells]` - Total number of positioned elements Note: Lines are not counted as an element.
Status of ongoing command sequence	Visualizes the current status of the command sequence in progress - `In execution` - `Paused` - `Aborted` Note: Only visible in execution mode.
Current cursor position	Current position of the mouse pointer in the command sequence grid. `[cell position of horizontal cells] x [cell position of vertical cells]`
Teaching position	Current position of the teaching cursor in the command sequence grid. Note: only visible if teaching is in progress.
Number of selected elements	Visualization of the elements selected in the command sequence grid: `n elements selected` Note: Only visible in editing and teaching mode.
Zoom level set	Drop-down list to select the zoom level for the display of the command sequence grid. Select the desired level in percent from a drop-down list. Default: `100 %`
Slider for zoom	Selection of zoom level for the display of the command sequence grid. The zoom level can be selected ether with the slider or with the **-** and **+** buttons. The zoom levels are identical to the drop-down list of Set zoom level

Selection of the dockable windows	Drop-down list for **List of command sequences**. Selection of whether the window shows the **List of command sequences** or is hidden. If the list of command sequences is already shown, this is visualized in the drop-down list with a tick.

Footer symbols

The footer of the command sequences grid uses the following symbols for visualization:

✕ Aborted

✓ Executed

👍 Ready for start

▶ In execution

🖥 Automatic mode

✎ Edit mode

🖐 Semi-automatic mode

▬ Idle

⏸ Paused/pause

⊡ Teaching

❗ Warning

11.1.4 Modes

The command sequence editor has several modes:

- Edit mode (à la page 60)
  Command sequences are configured and edited in this mode
- Execution mode (à la page 90)
  Mode for execution of the configured command sequences.
  The execution mode has two modes:
  - Automatic mode

- Semi-automatic mode

Command sequence - execution mode

The command sequence editor executes command sequences in Runtime. You can start any number of command sequences. The execution mode is for testing a command sequence but also to execute this. In addition in the execution 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 execution mode:

- Automatic mode (à la page 59)
  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 (à la page 59)
  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.

> **Informations**
>
> The fundamental command sequence process cannot be changed in execution 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.

11.1.5 Toolbar - Command Sequence Editor (edit mode)

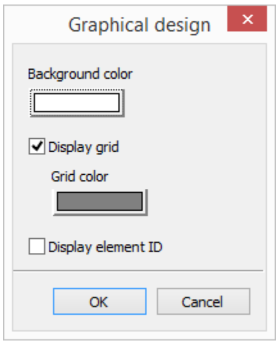
In edit mode of the command sequence editor, you can easily configure a command sequence in Runtime directly by means of drag&drop.

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 (à la page 63). 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 (à la page 97) 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 (à la page 72) and transition (à la page 68). Note: not available for step.
Replace step	Opens the dialog (à la page 67) 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 (à la page 68). It can be added to any allowed, free location via click.
Insert begin simultaneous sequence	Occupies the mouse cursor with a begin parallel branch (à la page 69). It can be added to any allowed, free location via click.
Insert end simultaneous sequence	Occupies the mouse cursor with an end parallel branch (à la page 69). It can be added to any allowed, free location via click.
Insert Begin branch	Occupies the mouse cursor with a begin branch (à la page 70). It can be added to any allowed, free location via click.
Insert End branch	Occupies the mouse cursor with an end branch (à la page 70). It can be added to any allowed, free location via click.
Insert switchgear allocation	Occupies the mouse pointer with a switchgear allocation (à la page 72). It can be added to any allowed, free location via click.
Insert jump target	Occupies the mouse cursor with a jump target (à la page 75). It can be added to any allowed, free location via click.
Start teaching	Starts teaching mode for the currently-selected command sequences. Not active if: ▶ Teaching mode is currently running for another command sequence. ▶ Project is not running in simulation mode.
Stop teaching	Ends teaching mode. Not active if teaching mode is active.
Switch command sequence to execution mode	Switches command sequence to execution mode (à la page 90).

Graphical design

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


Graphical design

Background color

☑ Display grid

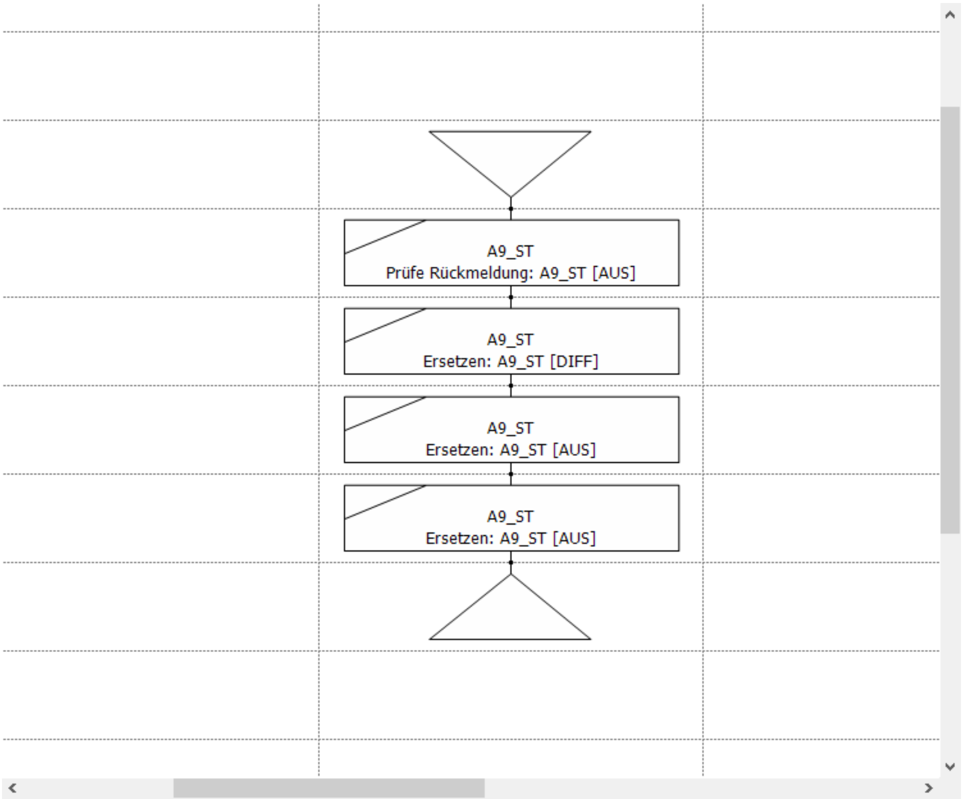
Grid color

□ Display 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.

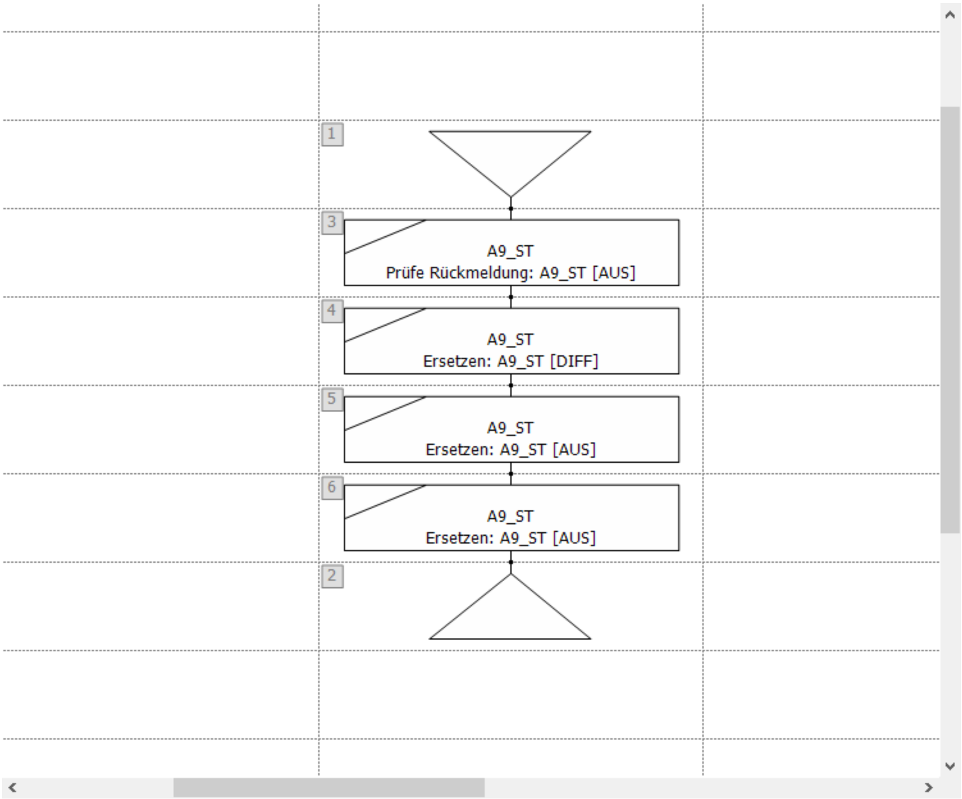
EXAMPLES

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 (à la page 67)
- Transition (à la page 68)
- Parallel branch (à la page 69)
- Branch (à la page 70)
- Switchgear allocation (à la page 72)
- Jump target (à la page 75)
- Lines (à la page 76)
- Teaching cursor (à la page 75)

START AND END ELEMENT

Each command sequence must have a start and end element.

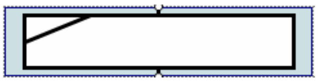
These two elements are automatically created when a command sequence is created and cannot be deleted from the project configuration. Even if you have configured the complete project configuration and deleted it, the start element and the end element are not affected by this deletion.

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 grid by means of drag&drop.
  Note: You can position the element several times.
- Clicking on the **Esc** button deactivates the drag&drop functionality.

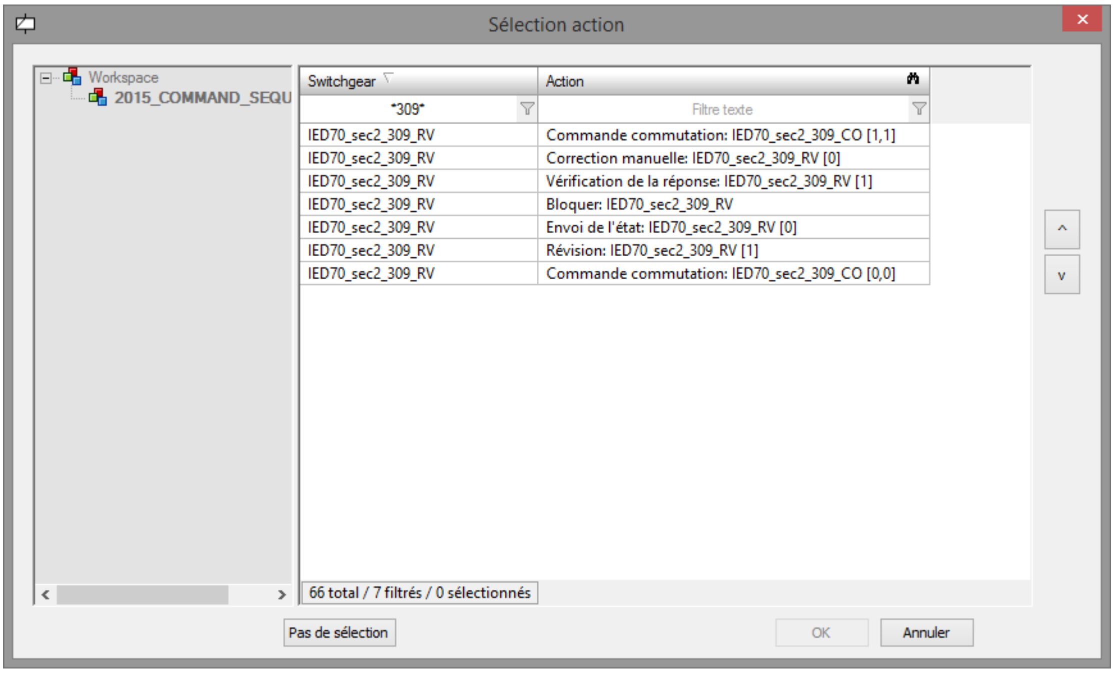
Step

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 **Action selection** 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



Switchgear	Action
309	Filtre texte
IED70_sec2_309_RV	Commande commutation: IED70_sec2_309_CO [1,1]
IED70_sec2_309_RV	Correction manuelle: IED70_sec2_309_RV [0]
IED70_sec2_309_RV	Vérification de la réponse: IED70_sec2_309_RV [1]
IED70_sec2_309_RV	Bloquer: IED70_sec2_309_RV
IED70_sec2_309_RV	Envoi de l'état: IED70_sec2_309_RV [0]
IED70_sec2_309_RV	Révision: IED70_sec2_309_RV [1]
IED70_sec2_309_RV	Commande commutation: IED70_sec2_309_CO [0,0]

66 total / 7 filtrés / 0 sélectionnés

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

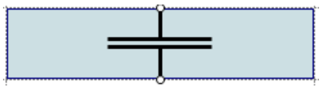
Note: The **OK** button is active if no action has been selected. Only the linking of an action is permitted for a step.

Informations

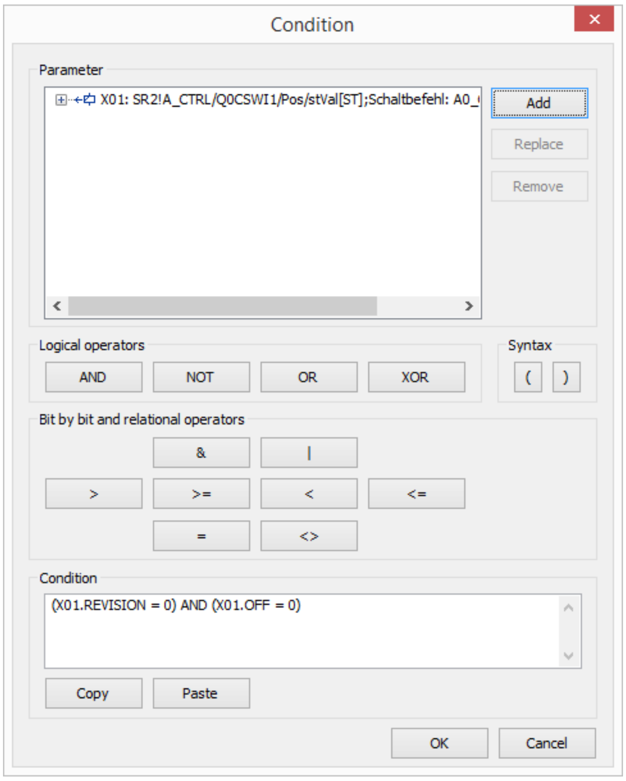
You can find further information on configuration of the command processing in the Energy Edition manual in the Command processing chapter.

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.



X01: SR2!A_CTRL/Q0CSWI1/Pos/stVal[ST];Schaltbefehl: A0_

(X01.REVISION = 0) AND (X01.OFF = 0)

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.

Informations

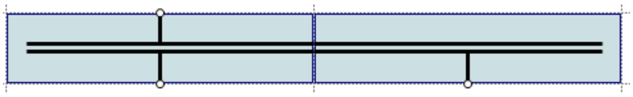
You can find further information on conditions in the Formula editor (à la page 78) chapter in this manual.

Parallel branches

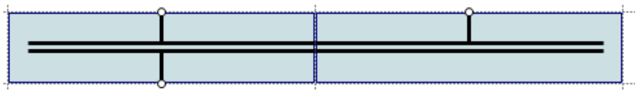
At the parallel branch an execution path parts into several execution paths which are executed in parallel during the process. For the activation of the different elements within a parallel branch you cannot define a certain order.

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:

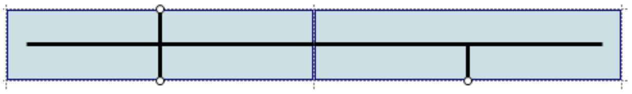
1. select the symbol **Insert begin parallel branch**
2. put the branch on the desired location
3. connect the input connection point with a output connection point of the preceding object
4. connect both output connection points with the desired following objects
5. close a parallel branch with object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

Branch

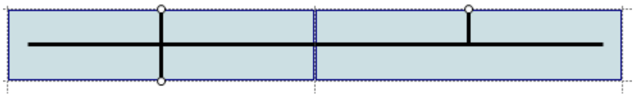
A branch offers the possibility to execute one of several possible paths. To do this, it is necessary that the first element at the start of a procedure path is a **transition**.

Note: A **Begin branch** can only be have a transition (à la page 68) 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 (à la page 69).

branches and parallel branches consist of:

- Single or 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.

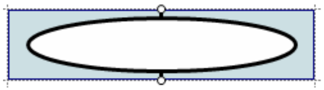
CHANGE SIZE

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.

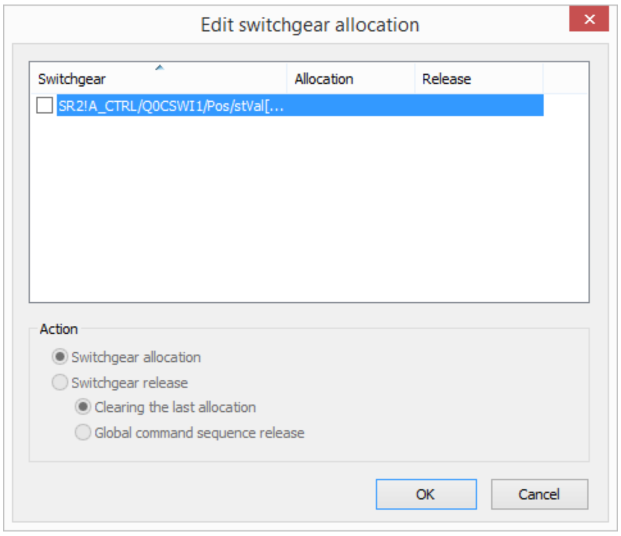


The **switchgear allocation** element of the module allocates one or more switching devices in Runtime:

- 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.


Edit switchgear allocation
Switchgear
Allocation
Release
☐ SR2!A_CTRL/Q0CSWI1/Pos/stVal[...
Action
● Switchgear allocation
○ Switchgear release
● Clearing the last allocation
○ Global command sequence release
OK
Cancel

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

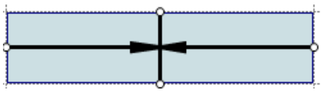
Parameters	Description
Switchgear allocation	If activated, the element is allocated to selected switchgear.
Switchgear release	If active, the element releases the selected switchgear.
Clearing the last allocation	If active, only the switchgear that was allocated in the last allocation is released. Only 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.

FERMER BOÎTE DE DIALOGUE

Paramètres	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.

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

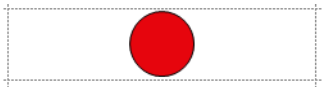
Jump targets consist of tree inputs and one output. At this the output is always at the bottom and the inputs are located at the top and the sides. You can connect any input connection points. A path which ends in a jump target must have started with a **Begin branch**. Otherwise the end is not reached.

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.

Teaching cursor

The **teaching cursor** element visualizes each position in the command sequences grid, according to which the recorded element is placed during the teaching process.



The teaching cursor is not directly available to select as an element. It is provided by activating the teaching mode automatically.

Note: The teaching process is not started without the teaching cursor being positioned.
If the recording is ended by clicking on the **Stop teaching** button, the teaching cursor is also hidden again.

POSITIONING

To position the teaching cursor in the command sequence grid:

1. Click (with simulation mode active) the **Start Teaching** button.

2. The teaching cursor is displayed and can be positioned in the command sequence grid.

NEW POSITIONING

The teaching cursor can be repositioned during an ongoing teaching process.

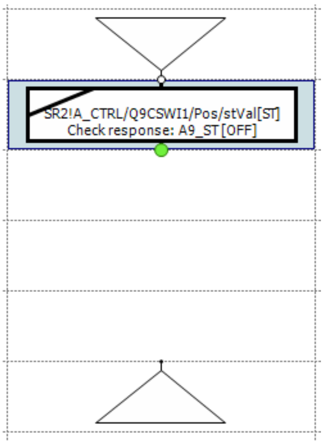
In the command sequence grid, move the teaching cursor with drag&drop and place the teaching cursor at the desired position in the grid. The steps applied from teaching are inserted into this new cursor position.

This repositioning can be repeated several times. As a result, different branches of a parallel branch can be configured with the teaching process.

Lines

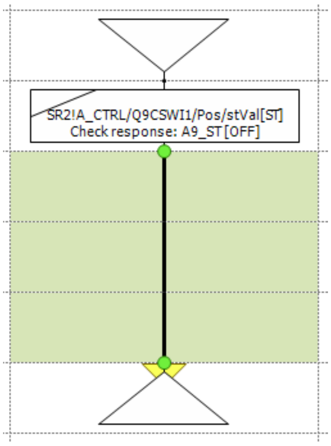
Lines connect elements via free connection points. To connect connection points with each other:

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



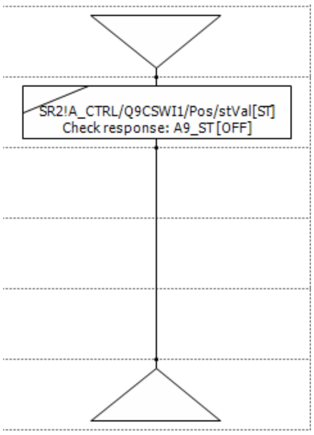
SR2!A_CTRL/Q9CSWI1/Pos/stVal[ST]
Check response: A9_ST [OFF]

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.



SR2!A_CTRL/Q9CSWI1/Pos/stVal[ST]
Check response: A9_ST [OFF]

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



SR2!A_CTRL/Q9CSWI1/Pos/stVal[ST]
Check response: A9_ST [OFF]

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 (à la page 97) 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.

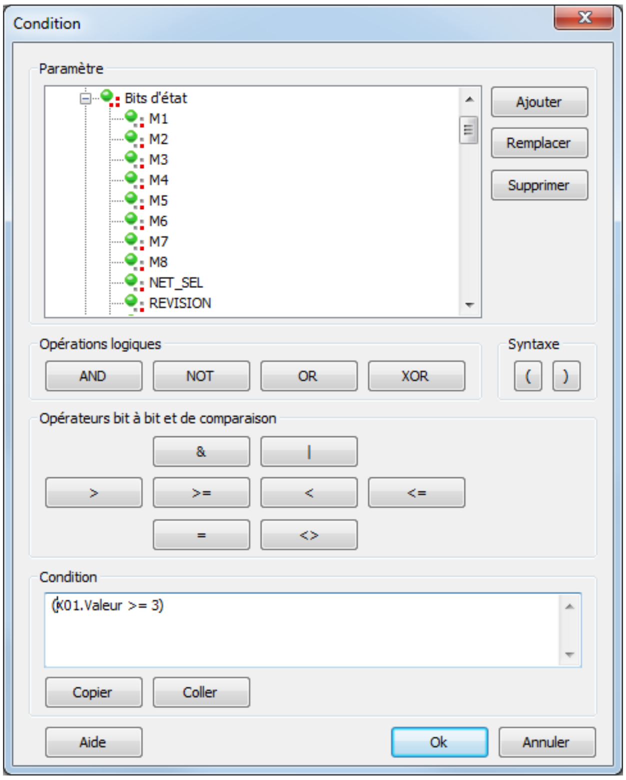
Note: If the step referenced in the formula is removed and a new step is added, the operands are reassigned in the case of operands. To do this, the same step must be reinserted. Parameters from a different phase are not automatically linked.

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



Condition: (K01.Valeur >= 3)

Parameters	Description
Parameter 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 indentifier at the beginning of the name is used for the formula.
Add	Opens the dialog for adding a parameter (à la page 83). 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 (à la page 83). - 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 than 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.

Informations

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 (à la page 84).
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` (à la page 87). 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 (à la page 87)

Info

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

Informations

Les formules comportant des valeurs X binaires et des liens au niveau des bits peuvent être utilisées avec 2 valeurs binaires maximum. Si d'autres valeurs sont requises, la liaison doit être établie sans valeurs X binaires.

Exemple :

X01.Value & X02.Value -> fonctionne

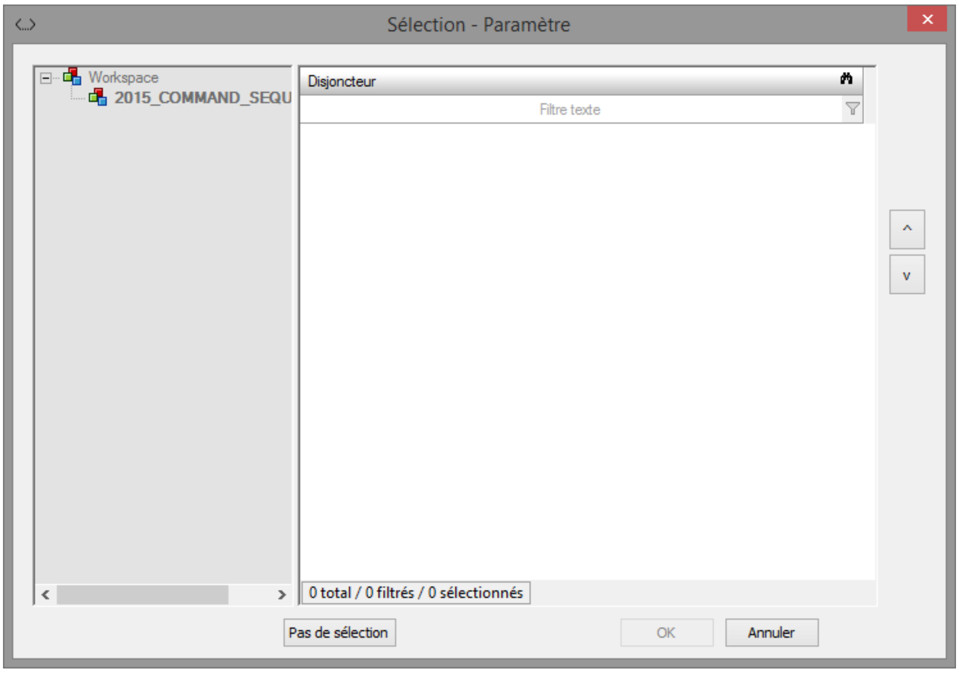
X01.Value & X02.Value & X03.Value -> ne fonctionne pas

Toutefois :

X01.00 AND X02.00 AND X03.00 AND X04.00 AND X05.00 -> fonctionne

Adding parameters

Clicking on the **Add** button in the formula editor (à la page 78) 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 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

Numéro de bit	Désignation abrégée	Nom long	Intitulé zenon Logic
0	M1	État utilisateur 1 ; pour le module Gestion de commande : type d'action Bloquer ; fonction Service Tracking (Main.chm::/IEC850.chm::/117281.htm) du driver IEC 850	_VSB_ST_M1
1	M2	État utilisateur 2	_VSB_ST_M2
2	M3	État utilisateur 3	_VSB_ST_M3
3	M4	État utilisateur 4	_VSB_ST_M4
4	M5	État utilisateur 5	_VSB_ST_M5
5	M6	État utilisateur 6	_VSB_ST_M6
6	M7	État utilisateur 7	_VSB_ST_M7
7	M8	État utilisateur 8	_VSB_ST_M8
8	NET_SEL	Sélectionné dans le réseau	_VSB_SELEC
9	REVISION	Révision	_VSB_REV
10	PROGRESS	En fonctionnement	_VSB_DIREC
11	TIMEOUT	Dépassement dans le Runtime	_VSB_RTE
12	MAN_VAL	Valeur manuelle	_VSB_MVALUE
13	M14	État utilisateur 14	_VSB_ST_14
14	M15	État utilisateur 15	_VSB_ST_15
15	M16	État utilisateur 16	_VSB_ST_16
16	GI	Requête générale	_VSB_GR
17	SPONT	Spontané	_VSB_SPONT
18	INVALID	Invalide	_VSB_I_BIT
19	T_CHG_A	Annonce heure été/hiver	_VSB_SUWI
20	OFF	Désactivé	_VSB_N_UPD
21	T_EXTERN	Temps réel externe	_VSB_RT_E
22	T_INTERN	Horodatage temps réel interne	_VSB_RT_I
23	N_SORTAB	Non triable	_VSB_NSORT
24	FM_TR	MD_TR;Message de défaut du transfo	_VSB_DM_TR
25	RM_TR	Message de marche du	_VSB_RM_TR

		transformateur	
26	INFO	Informations de la variable	_VSB_INFO
27	ALT_VAL	Valeur de remplacement Si aucune valeur n'a été transférée, la valeur de remplacement définie est utilisée ; dans le cas contraire, la dernière valeur valide est utilisée.	_VSB_AVALUE
28	RES28	Réservé à une utilisation interne (clignotement d'alarme)	_VSB_RES28
29	N_UPDATE	Non mis à jour	_VSB_ACTUAL
30	T_STD	Heure standard	_VSB_WINTER
31	RES31	Réservé à une utilisation interne (clignotement d'alarme)	_VSB_RES31
32	COT0	Cause de transmission bit 1	_VSB_TCB0
33	COT1	Cause de transmission bit 2	_VSB_TCB1
34	COT2	Cause de transmission bit 3	_VSB_TCB2
35	COT3	Cause de transmission bit 4	_VSB_TCB3
36	COT4	Cause de transmission bit 5	_VSB_TCB4
37	COT5	Cause de transmission bit 6	_VSB_TCB5
38	N_CONF	Confirmation négative de la commande par l'appareil (IEC 60870 [P/N])	_VSB_PN_BIT
39	TEST	Bit de test (IEC60870 [T])	_VSB_T_BIT
40	WR_ACK	Écriture reconnue	_VSB_WR_ACK
41	WR_SUC	Écriture réussie	_VSB_WR_SUC
42	NORM	NORM;État Normal	_VSB_NORM
43	N_NORM	État déviation normale	_VSB_ABNORM
44	BL_870	État IEC 60870 : bloqué	_VSB_BL_BIT
45	SB_870	État IEC 60870 : substitué	_VSB_SP_BIT
46	NT_870	État IEC 60870 : non topique	_VSB_NT_BIT
47	OV_870	État IEC 60870 : dépassement	_VSB_OV_BIT
48	SE_870	État IEC 60870 : sélectionné	_VSB_SE_BIT
49	T_INVAL	Temps invalide	non défini
50	CB_TRIP	Déclenchement disjoncteur détecté	non défini
51	CB_TR_I	Détection déclenchement disjoncteur inactive	non défini

52	OR_DRV	Valeur hors plage	non défini
53	RES53	réservé	non défini
54	RES54	réservé	non défini
55	RES55	réservé	non défini
56	RES56	réservé	non défini
57	RES57	réservé	non défini
58	RES58	réservé	non défini
59	RES59	réservé	non défini
60	RES60	réservé	non défini
61	RES61	réservé	non défini
62	RES62	réservé	non défini
63	RES63	réservé	non défini

Informations

Dans les formules, tous les bits d'état sont disponibles. La disponibilité peut être réduite dans le cadre d'autres utilisations.

Pour plus de détails concernant la gestion des états, reportez-vous au chapitre Gestion d'états.

Logical operators

Liens logiques : seule la valeur logique '0' sera recherchée dans les variables ; si la valeur n'est pas égale à '0', elle sera considérée comme égale à '1'.

Contrairement aux formules bit, la portée technique peut être modifiée au moyen d'un facteur d'étirement -> (différent de '0' ou '1').

Opérateur	Signification
AND	ET logique
NOT	Négation
OR	OU logique
XOR	OU EXCLUSIF logique

Les opérateurs possèdent la priorité suivante dans le calcul de la formule :

Priorité	Opérateur
1	& (opérateur des formules bit)
2	NOT
3	AND
4	XOR/OR

Informations concernant

Jusqu'à 99 variables peuvent être liées dans une formule. X01 à X99.

Informations concernant

Les bits d'état NORM et N_NORM sont uniquement disponibles dans l'éditeur de formules et ne peuvent pas être configurés par l'intermédiaire de l'état.

Bit formulas

Les formules Bit comportent uniquement un état logique haut ou bas. Contrairement aux formules logiques, la valeur brute est déjà prédéfinie (**0**,1).

Opérateur	Description
&	AND
|	OR

Example: ORing bitwise

Vous voulez savoir si l'un des bits d'état utilisateur 1 à 8 (M1 ... M8) de la variable X01 est défini :

FORMULE CLASSIQUE :

X01.M1 OR X01.M2 OR X01.M3 OR X01.M4 OR X01.M5 OR X01.M6 OR X01.M7 OR X01.M8.
Cette requête peut être considérablement simplifiée en utilisant la condition logique OR sur l'état général.

INTERROGATION OR LOGIQUE :

X01.Status & 0xFF

La constante peut être saisie au format hexadécimal, comme ci-dessus.

0xFF en décimal vaut 256, et correspond aux huit premiers bits d'état (en binaire, 11111111). Si l'un de ces bits est défini sur 1, le résultat de l'application de la condition logique ORing au niveau des bits est 1 (True) ; dans le cas contraire, le résultat est 0 (False).

Si, par exemple, tous les bits d'états utilisateur sont à tester sauf le bit M7, la formule binaire serait : 10111111. Le bit 7 est sans intérêt, et est donc défini sur 0. Ceci correspond à 0xBF en hexadécimal. L'expression de la formule est alors : **X01.Status & 0xBF**.

A la place d'une comparaison de bits avec OR à l'aide d'une constante, la valeur peut également être comparée directement à un nombre décimal. Si la comparaison est fausse, la valeur binaire sera égale à 0 (False) ; dans le cas contraire, elle prendra la valeur 1 (True).

Exemple :

Si vous voulez savoir si la valeur est égale à la constante 202 : La formule est alors :

X01.value = 202

Si la valeur est égale à la constante 202, le résultat de la formule est 1 (True) ; dans le cas contraire, le résultat est 0 (False).

Remarque : avec le caractère OR (|), la condition logique OR au niveau des bits fonctionne de la même manière que dans cet exemple.

Comparison operators

Les opérateurs de comparaison permettent de comparer directement deux valeurs numériques. Le résultat des comparaisons est une valeur binaire. "0" si la condition est fausse et "1" si la condition est vraie.

Opérateur	Description
<	Inférieur
>	supérieur
<=	Inférieur ou égal
>=	supérieur ou égal
=	Égal
<>	Différent de

À gauche et à droite de l'opérateur de comparaison doivent se trouver des valeurs complètes ou des états complets ; des bits simples ne peuvent pas être utilisés.

A droite d'un opérateur de comparaison, vous pouvez également utiliser une constante
Ces constantes sont saisies sous forme de valeurs hexadécimales ou de valeurs décimales dans l'élément combiné. Les chiffres hexadécimaux sont automatiquement convertis en valeurs décimales lorsque vous cliquez sur **OK** (par exemple, 0x64 égale 100 en chiffres décimaux).

> **Exemple**
>
> *X01.valeur >= X02.valeur*
> *Le résultat est 1, si la valeur de X01 est supérieure ou égale à la valeur de X02*
>
> *X01.valeur = 0x64*
> *Le résultat est 1, si la valeur de X01 est exactement égale à la valeur numérique 100 (= hex 0x64)*
>
> *(X01.valeur = 0x64) OR (X01.valeur = 0x65)*
> *Le résultat est 1, si la valeur de X01 est exactement égale à la valeur numérique 100 ou 101 (= hex 0x64 ou hex 0x65)*

Examples for formulas

LIAISON LOGIQUE AND SIMPLE ENTRE DEUX VALEURS DE BITS

> **Exemple**
>
> *Formule : X01.03 AND X02.03*

Cette formule vaut TRUE si le **bit 3** de la variable `1` et le **bit 3** de la variable `2` sont tous deux égaux à `1`.

COMPARAISON D'UNE VALEUR ANALOGIQUE OU DE L'ÉTAT D'UNE VARIABLE

> **Exemple**
>
> *(X01.Value > X02.Value)*

COMPARAISON DE VALEURS ANALOGIQUES SUR UNE BASE LOGIQUE

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

COMPARAISON À L'AIDE DE BITS DE VALEUR ET DE BITS D'ÉTAT

Exemple

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

COMPARAISON D'UNE VALEUR AVEC UNE VALEUR DÉCIMALE OU HEXADÉCIMALE

Exemple

Formule : (X01.Value = 111)

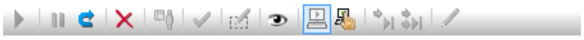
Formule : (X01.Value = 0x6F)

Si une valeur hexadécimale est utilisée, cette dernière est convertie en valeur décimale lorsque vous cliquez sur **OK**. Si une valeur décimale est saisie et confirmée, la valeur reste affichée au format décimal après la réouverture.

Informations concernant

Il n'est pas possible d'utiliser une virgule ou un point lors de la saisie d'une valeur.

11.1.6 Toolbar - Command sequence editor (execution mode)



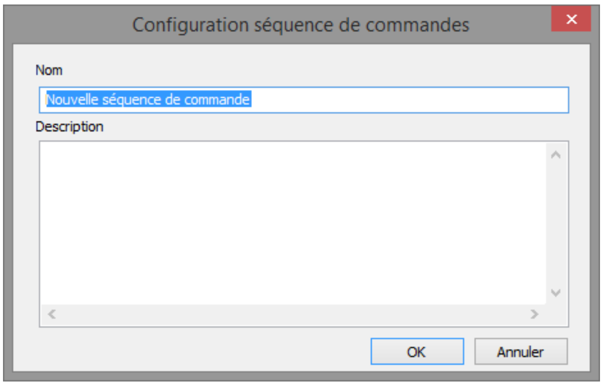
Parameters	Description
Start command sequence	Starts command sequence Note: not available during redundancy switching.
Pause command sequence	Stops current command sequence and pauses it.
Continue command sequence	Continues paused command sequence. Only active if the command sequence was previously paused 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 (à la page 72) and transition (à la page 68). No changes can be made in execution 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 (à la page 63). 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` (à la page 59).
Switch to semi-automatic mode	Switches execution of the switching step to `semi-automatic mode` (à la page 59).
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.

11.2 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 (à la page 34).



Nouvelle séquence de commande

Parameters	Description
Name	Name of the new command sequence. The name must not contain a question mark (**?**), a semicolon, 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.

FERMER BOÎTE DE DIALOGUE

Paramètres	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.

11.3 Tooltips

Tool tips in the command sequence editor visualize the respective status of a step and provide further information via the respective status (à la page 94).

DISPLAY OF THE TOOL TIP:

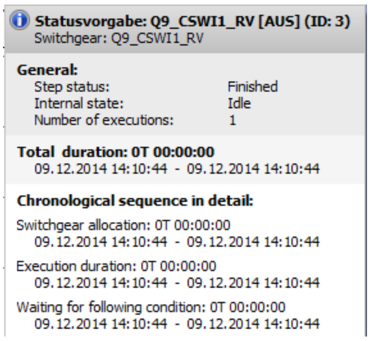
To have a tool tip displayed, go to the respective step in the command sequence grid. The tool tip appears automatically when the mouse is positioned over the step.

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



```
Statusvorgabe: Q9_CSWI1_RV [AUS] (ID: 3)
   Switchgear: Q9_CSWI1_RV

General:
   Step status:                  Finished
   Internal state:               Idle
   Number of executions:         1

Total  duration: 0T 00:00:00
   09.12.2014 14:10:44  -  09.12.2014 14:10:44

Chronological sequence in detail:
Switchgear allocation: 0T 00:00:00
   09.12.2014 14:10:44  -  09.12.2014 14:10:44
Execution duration: 0T 00:00:00
   09.12.2014 14:10:44  -  09.12.2014 14:10:44
Waiting for following condition: 0T 00:00:00
   09.12.2014 14:10:44  -  09.12.2014 14:10:44
```

11.4 Execution state

The following states are possible:

State	Description
`Idle`	The command sequence is in idle state.
`In execution`	When starting a command sequence, it changes to `running` status.
`Skipped`	A step was skipped due to the project configuration (à la page 28).
`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 command sequence, 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

State	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.
Pause	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 command sequence is deleted.

11.5 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 (à la page 112).
🔴	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 (à la page 94) of **steps**, **transitions** and **End parallel branch** is visualized in color:

Status	Color
Idle:	white
In execution:	green
Skipped:	petrol
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.

11.6 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 execution mode.

To validate a command sequence, click on the corresponding symbol in the toolbar 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:

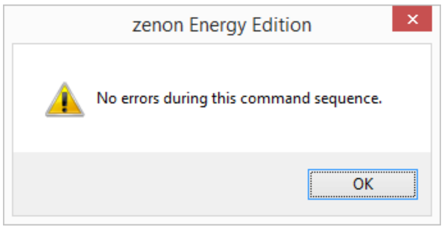
- Syntax (all lines connected, processable from begin to end, etc.)
- Variables
- Data Types

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.*

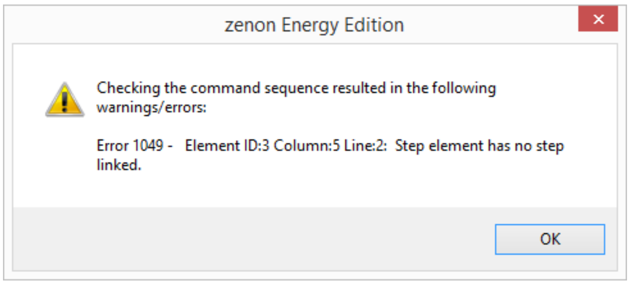
DIALOG: VALIDATION OK



zenon Energy Edition

No errors during this command sequence.

DIALOG: VALIDATION WITH ERRORS



zenon Energy Edition

Checking the command sequence resulted in the following warnings/errors:

Error 1049 - Element ID:3 Column:5 Line:2: Step element has no step linked.

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

11.7 Teaching

Configured command processes can be recorded in the zenon Runtime simulation with the help of teaching. As a result of this process, corresponding command sequences are configured in the simulation image in Runtime during the teaching process.

The command sequence configuration created this way can still be changed manually and imported in real time by importing the simulation image created as a result.

All steps are carried out in zenon Runtime. Additional configuration in for teaching in the zenon Editor is not necessary.

11.7.1 Teaching process

Command actions are processed in a project simulation during an active teaching process and applied in the command sequence selected for teaching. The command action is incorporated as a step element in the command sequence grid. Elements can also be inserted manually in the command sequence grid during teaching. The teaching process is not interrupted as a result.

In order to be able to execute command actions from different process images, the teaching process also remains active when the command sequences editor is closed.

During this process, it is possible to switch between command images and desired zenon screens in Runtime. It is thus also possible to apply command actions from different zenon screens in the command sequence. A requirement for this is that there is at least one command action included in the respective zenon screen.

Teaching is always carried out in the project simulation. As a result, it is ensured that ongoing real-time operation is not affected or that damage is caused by incorrect project configurations. The command sequence can also be edited and changed after the teaching process. An element can still be added manually and processes can be rearranged.

For a command sequence, the teaching process can be repeatedly started and stopped in the simulated Runtime environment.

The command sequences edited in this way can then be imported into the **list of command sequences** (à la page 40) in real-time operation with the **Import command sequence from simulation image** (à la page 49) button.

TEACHING IN THE COMMAND SEQUENCE GRID

The teaching cursor can be positioned in any free cell in the command sequence grid by means of drag&drop. Repositioning of the teaching cursor is possible at any time. For example, for parallel branches, the individual branches can be taught consecutively.

During the teaching process, for each command action executed, the corresponding step is inserted into the cell of the command sequence grid on which the teaching cursor is currently located. Once a step has been inserted, the teaching cursor is moved down one cell.

If the new teaching cursor position is already occupied by an existing command sequence project configuration, all elements below this in the grid are moved down one line. If existing lines are separated as a result, these are automatically connected again. As a result, it is ensured that a valid command sequence that already exists remains free of errors through the teaching process.

Expansion of the grid is possible up to a total of 1,000 lines. The teaching is canceled after this limit has been reached. The user is informed of this with a dialog.

11.7.2 Dialog when teaching is canceled

Certain events in zenon Runtime lead to an ongoing teaching process being canceled. This cancellation of the teaching process is visualized with warning dialogs. In addition, CEL entries and LOG entries are created for the respective events.

The teaching process is canceled:

- When the project simulation is stopped in Runtime.
- When the **Stop teaching** button is clicked on.
- When the mode of the taught command sequence is switched from `edit mode` into `execution mode`.
- During an XML export of the taught command sequence.
  Note: Caution, no warning dialog appears here. The teaching is stopped automatically. The previously-taught elements are retained however.
- With XML import, if the taught command sequence is already present in the XML import file and `Overwrite existing command sequence` is selected for the import options. The previously-taught elements are lost!
- In the event of changes to the taught command sequence:
  - Save
  - Delete
  - Rename

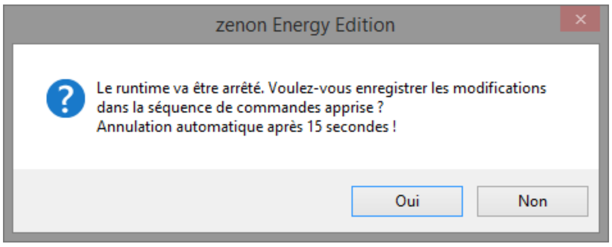
In doing so, no distinction is made on how the command sequence is changes: for example, by means of keyboard shortcut, clicking on a corresponding symbol, selection from a context menu, etc.

DIALOGS DURING TEACHING

When the teaching process is canceled, the user is notified of this with warning dialogs. Different dialogs are shown depending on the cause of cancellation.

CLOSING ZENON RUNTIME

If Runtime is closed during an active teaching process, the user is notified of this with a dialog:



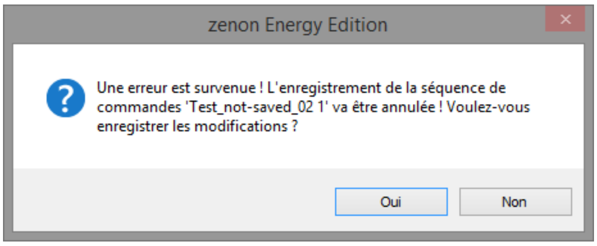
Le runtime va être arrêté. Voulez-vous enregistrer les modifications dans la séquence de commandes apprise ?
Annulation automatique après 15 secondes !

Parameters	Description
Yes	The changes are saved in the current command sequence to be taught. Runtime is closed.
No	The Runtime will be closed. The new elements of the command sequence are not saved.

If no button is clicked on, Runtime is automatically closed once 15 seconds have elapsed. The current command sequence is not saved.

NO AVAILABLE LINE IN THE COMMAND SEQUENCE GRID

The command sequence grid is expanded by one line during the teaching process if necessary. If the maximum number of 1,000 lines has been reached, the teaching process is canceled and the following dialog is displayed:

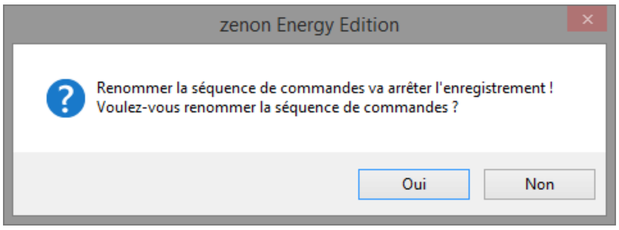


Une erreur est survenue ! L'enregistrement de la séquence de commandes 'Test_not-saved_02 1' va être annulée ! Voulez-vous enregistrer les modifications ?

Parameters	Description
Yes	Ends the active teaching process and saves the project configuration of the command sequence.
No	Ends the active teaching process. Caution: All new elements since the last save are lost.

RENAMING THE COMMAND SEQUENCE

If a command sequence is renamed during an active teaching process, the teaching is canceled. The following dialog appears:



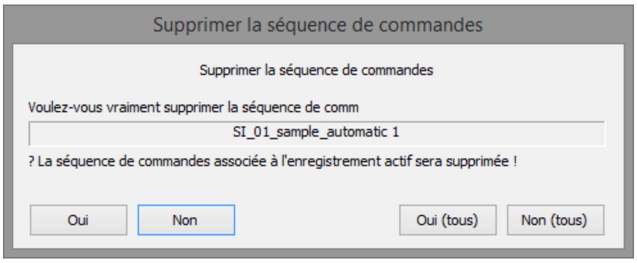
zenon Energy Edition

Renommer la séquence de commandes va arrêter l'enregistrement !
Voulez-vous renommer la séquence de commandes ?

Parameters	Description
Yes	Ends the active teaching process. The command sequence is saved with the current content for the command sequence. A new teaching process is possible.
No	The command sequence is not renamed. The current teaching process is continued. Changes that have previously been made are not saved.

DELETING THE COMMAND SEQUENCE

If a command sequence is deleted during an active teaching process, the following dialog is displayed before deletion:



Supprimer la séquence de commandes

Supprimer la séquence de commandes

Voulez-vous vraiment supprimer la séquence de comm

SI_01_sample_automatic 1

? La séquence de commandes associée à l'enregistrement actif sera supprimée !

Parameters	**Description**
`[Name of the command sequence]`	Name of the command sequence that is to be deleted. It is always only the current command sequence name that is displayed. The next command sequence to be deleted is shown after the **Yes** or **No** button is clicked on.
Yes	Ends the active teaching process. The command sequence is saved with the current content. A new teaching process is possible.
No	The command sequence is not renamed. The current teaching process is continued. Changes that have previously been made are not saved.
Yes all	Deletes all selected command sequences without requesting confirmation. Note: Not active if only one command sequence has been selected for deletion.
No all	Does not delete any of the selected command sequences. The teaching process for the command sequence to be taught continues to be active. Note: Not active if only one command sequence has been selected for deletion.

11.7.3 Engineering

In order to be able to use teaching in zenon Runtime, process screens with a corresponding (command input) configuration in the zenon Editor are required.

In addition, carry out the following steps:

ENGINEERING IN THE EDITOR

- Create a screen of type `Command Sequencer`
- Insert the `Command sequence editor` control element template into the screen.
- Create a screen switching function for the configured screen.
- Create an **activate/deactivate project simulation** function.
- Link the created functions in the project.

RUNTIME

1. Start the zenon Runtime.
2. Start the project simulation.
3. Change to the Command Sequencer screen
   a) Create a new command sequence or select an existing command sequence for editing.
   b) Click on the **Start teaching** button in the command sequence editor.
   c) Position the **teaching cursor** element in the command sequence grid.
4. Switch to any desired process screen.
5. Carry out the command actions.
   The corresponding step to the command sequence is added in the command sequence grid. This also happens if the command sequence image is not displayed in Runtime.
6. If necessary, edit the command sequence configuration. To do this, switch to the command sequencer screen:
   a) Taught steps can be repositioned in the command sequence grid by means of drag&drop.
   b) The teaching cursor can be repositioned in the command sequence grid for another teaching process by means of drag&drop.
   c) To do this, continue the teaching process with step 4.
7. End the teaching process by clicking on the **Stop teaching** button.
8. Correct the end element at the end of the command sequence configuration.
   Note: An empty line always remains before the end element by ending the teaching process. Move the end element up one position or drag a line through the empty cell that has been created in order to complete the command sequence configuration correctly.
9. Check your command sequence configuration by clicking on the **Check command sequence for errors** button.
10. End the Runtime simulation.
11. Switch to the command sequence editor
12. In the **List of command sequences** window, click on the **Import command sequence from simulation image** (à la page 49) symbol.
13. Select, in the **import options** dialog, the desired command sequence and accept this by clicking on the **OK** button in the **list of command sequences**.

11.8 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 not 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 (à la page 97) 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.

11.9 CEL entries

Processes in zenon Runtime are supported and visualized by entries in the Chronological Event List.

The following entries are written to the CEL:

- Command sequence created:
  `Command sequence [command sequence name] created`

- Command sequence duplicated:
  `Command sequence duplicated. New: [command sequence name] source: [command sequence name]`
- Command sequence renamed:
  `Command sequence renamed. New: [command sequence name] old: [command sequence name]`
- Command sequence completed:
  `Command sequence [command sequence name] completed`
- Command sequence canceled:
  `Command sequence [command sequence name] canceled`
- Manual script executed:
  `Manual script executed. Command sequence [command sequence name]; previous element ID: [Element ID] (column [column number in the command sequence grid] – line [line number in the command sequence grid])`
- Manual jump carried out:
  `Manual jump carried out. Command sequence [command sequence name]; start: Subsequent element ID: [Element ID] (column [column number in the command sequence grid] – line [line number in the command sequence grid]) – target: Previous element ID: [Element ID] (column [column number in the command sequence grid] – line [line number in the command sequence grid])`
- Skip step:
  `The action  [step name] was skipped.`

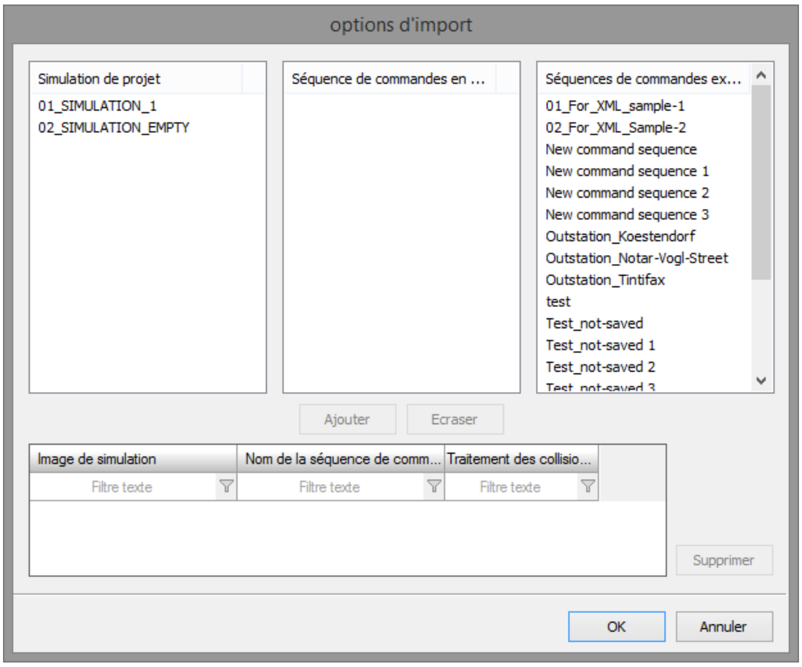
Informations

You can find further information on CEL in the Chronological Event List manual.

12. Command sequences and simulation mode

12.1 Import command sequence(s) from simulation image

In this dialog, you arrange configured command sequences from one or more simulation images for import into zenon Runtime.



Simulation de projet:

01_SIMULATION_1
02_SIMULATION_EMPTY

Séquences de commandes ex...:

01_For_XML_sample-1
02_For_XML_Sample-2
New command sequence
New command sequence 1
New command sequence 2
New command sequence 3
Outstation_Koestendorf
Outstation_Notar-Vogl-Street
Outstation_Tintifax
test
Test_not-saved
Test_not-saved 1
Test_not-saved 2
Test_not-saved 3

Parameters	Description
Project simulation	List of all existing project simulation images. Only one simulation image can be selected at a time.
Command sequences in project simulation	List of all configured command sequences from the selected project simulation. Multiple selection is possible.
Existing command sequences	Pre-existing command sequences in the zenon project that is currently running.
Add	Adds selected command sequences from the **Command sequences in project simulation** list to the `List of command sequences to be applied`. Only active if at least one command sequence from a simulation image has been selected.
Overwrite	Adds a command sequence from the **Command sequences in project simulation** list to the `List of command sequences to be applied`. This command sequence overwrites the existing command sequence in the command sequences editor in Runtime! This button is only active if just one command sequence has been selected and the selected command sequence is already in the **existing command sequences** list. The command sequence is thus already present in the command sequences editor and is overwritten. With multiple selection of command sequences from a project simulation, the button is then grayed out if the selection contains a command sequence that is not yet present in the list of **existing command sequences**. A new command sequence is created when transferring to Runtime. This new project configuration is given a serial number in the naming.
`List of the command sequences to be transferred.`	Assigned command sequences that are applied in the current Runtime environment. - `Simulation image:` Name of the simulation image from which the command sequence comes. - `Command sequence name:` Name of the command sequence as saved in the simulation image. - `Conflict handling`: The type of conflict handling depends on how the respective command sequence is

	transferred to the list. Depending on the button used (**Add** or **Overwrite**), the conflict handling is prescribed and cannot be changed. Note: List can be sorted and filtered.
Remove	Removes highlighted command sequence from the `List of command sequences to be applied`. Multiple selection is possible. A new assignment from **Command sequences in project simulation** is possible.
OK	Closes the dialog and applies project configurations from the `list of command sequences to be applied` in the command sequence editor.
Cancel	Annule toutes les modifications et ferme la boîte de dialogue.

If the **Command Sequencer** module is operated in the zenon network, the following rules are applicable:

- If the dialog is called up, the **project simulation** list is filled with the simulation images from the server.
  If a server is lost, the dialog with an empty **project simulation** list is called up.
- The **command sequences in project simulation** list is also filled with data from the server on the client. The list is empty if the server is lost.

12.2 Creating a simulation image

In order to be able to create a simulation image in zenon Runtime, carry out the following steps in the zenon Editor:

1. Create a new function:

   In the tool bar or in the context menu of the **Functions** node, select the **New function** command.
   The dialog to select a function is opened.
2. Select, in the dialog from the **Applications** group, the **Activate/deactivate project simulation** function.
   The dialog to configure the project simulation is opened.
3. Name the function.
4. Link the function to a button.

OPERATION IN RUNTIME

If the dialog is not offered in Runtime, please note that each restart of a simulation overwrites the previous simulation image!

> **Informations**
>
> *You can find more information in the project simulation manual.*
>
> *The current status is read with the system driver variable* ***[system information] Runtime status (simulation)***

13. Command sequences in the zenon network

The **Command Sequencer** module is also available in the zenon network. Command sequences are always executed on the Server in the process. Each client can execute and administer command sequences.

If a computer works as a Client in the network, all changes to the command sequence(s) are transferred to/from the **Primary Server**. If there is a **Standby Server** in the current network topology, all command sequences are synchronized by the **Primary Server**. The **Standby Server** synchronizes itself automatically.

DISPLAY OF DIALOGS

If a command sequence is started by means of a function or button in the command sequence editor, dialogs are always called up on the computer on which the command sequence was started.

These are:

- Command input screens
- Error message

REDUNDANCY

The **Command Sequencer** module supports, from zenon version 7.50, the redundant zenon network.

The following redundancy types are supported:

- Rated network
- Non-dominant network
- Dominant network

ZENON NETWORK

You can find further information on the configuration of redundancy in the Network manual.

- Types of redundancy
  - Redundancy in a rated network
  - Redundancy in a non-dominant network
- Authorization in the network
- Functions in the network

13.1 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 there 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.

13.2 Simulation images in network projects

If the **Command Sequencer** module is operated in the zenon network, the following rules are applicable:

- If the dialog is called up, the **project simulation** list is filled with the simulation images from the server.
  If a server is lost, the dialog with an empty **project simulation** list is called up.
- The **command sequences in project simulation** list is also filled with data from the server on the client. The list is empty if the server is lost.

13.3 Behavior in the zenon network

The network concept of the **Command Sequencer module** works according to the following principle:

- The command sequences can be configured on the client, as well as the **Server** or on the **Standby Server**.
- The configured command sequences are administered on the **Primary Server** and distributed to the clients.
- The command sequences can be operated on both the client and the **Server**.
- The command sequences are always executed on the **Primary Server**.
- With redundancy switching, the command sequences are canceled.
  These can be restarted manually on the new **Primary Server**.

ROLE SWITCH BETWEEN SERVER 1 AND SERVER 2

- Redundancy switching is delayed until all active command sequences have been completed.
- The start of command sequences is blocked during a redundancy switching. The buttons on the client are grayed out in this time.
  - This redundancy switching can be planned in a **Rated network**.
  - In a **dominant network** or a **non-dominant network**, redundancy switching is carried out when the **Primary Server** fails.
- It is possible to start command sequences again once the switch has been carried out or if the switch has been completed.
  An entry is written to the CEL in this case.
- CEL messages are written for the following events:
  - Start of a command sequence on the server is blocked.
  - If a command sequence in the dominant network is to be started on the **Serveur 2** .
  - The command sequence cannot be started because there is currently a redundancy switch pending.
  - Note: No CEL message is generated if an incorrect command sequence is started.

LOG ENTRY

Entry	Level	Description
`The sequence (mrid:<id1>, crid:<id2>)<name> could not be started, because a redundancy switch is pending.`	ERRORS	The command sequence cannot be started because there is a redundancy switch pending.

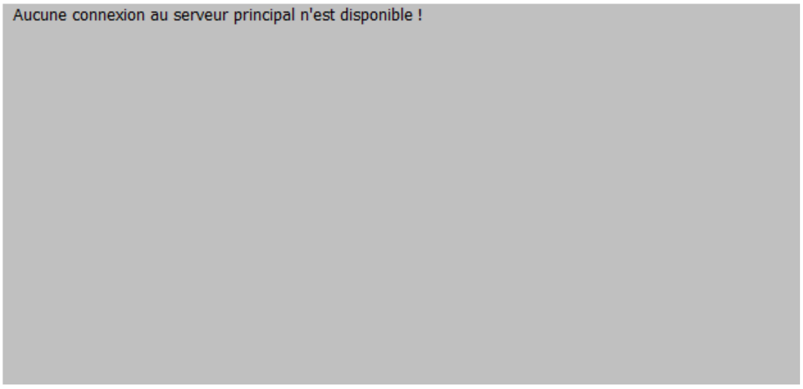
SPECIAL CASE: TWO SERVERS IN THE NETWORK

In the event that, when switching the **Primary Server**, there are still command sequences running on the "new" **Standby Server**, these are canceled on the Standby. This can only occur if both servers in the network were no longer connected (due to a network failure for example) and are now connected again. In this case, the change to the command sequence is not transferred to the Primary Server.

This means that, if there is a connection and command sequences that are now canceled on the Standby have already been opened on the Primary Server, these continue to be considered as running. It can only be restarted again once this command sequence has been closed on **Serveur 1** and **Serveur 2**.

NO CONNECTION TO SERVER AND STANDBY

If the command sequence screen is opened on the Client when neither **Serveur 1** nor **Serveur 2** are contactable, the command sequence editor on the Client is not available. The command sequence editor remains empty. An error text is displayed in zenon Runtime instead of the command sequence image.



Aucune connexion au serveur principal n'est disponible !

ERROR DIALOG

If a command sequence cannot be started, the following error dialog is called up:

14. 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 toolbar, 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.

15. Structure of the XML file for command sequences

Please keep in mind:

- Changes to the XML file are for experts only and are not generally recommended.
- An invalid XML file can lead to problems during XML import. The import can fail as a result.
- Due to the fact that both the **Command Sequencer** module and the **Batch Control** module use a joint XML structure, content and illustrations are the same as the terminology of the Batch Control module. If these entries are missing in the XML file, the content is also invalid for command sequences.

> **Informations**
>
> This documentation of the XML file describes the elements "from top to bottom" according to how they occur in the file. Substructures contained therein are each described in detail in their own areas.

Main nodes

The first level of the XML structure:

Parameters	Description
xml	XML declaration with XML version and character coding: `version="1.0" encoding="utf-16"`
Subject	General subject of the XML file

SUBJECT

The **Subject** node contains general information on the XML file.

Parameters	Description
ShortName	(prescribed value) `zenOn(R) exported project`
MainVersion	Version of zenon from which the XML export was carried out. (prescribed value) `7500`
Apartment	Please note the detailed description for the content of this node.

APARTMENT

The **Apartment** node represents content for exported command sequence project configurations.

Parameters	Description
ShortName	(prescribed value) `zenOn(R) command sequencer`
Version	Version of zenon from which the XML export was carried out. (prescribed value) `7500`
CommandSequence	Node for project configuration content of a command sequence. Each individual exported command sequence configuration is represented with its own `CommandSequence` node.

COMMANDSEQUENCE

The node contains general information on the configuration of a command sequence.

Parameters	Description
MrId	ID of the command sequence. This ID must be unique for each command sequence and must not be issued twice.
MrName	Name of the command sequence. Corresponds to the **Name** input field in the **Command sequence configuration** dialog.
MrDescription	Description of the command sequence. Corresponds to the **Description** input field in the **Command sequence configuration** dialog.
MrVersion	Version of command sequence. 1 (prescribed value) Default setting that cannot be changed.
MrSourceVersion	Original version of the command sequence. 0 (prescribed value) Default setting that cannot be changed.
MrStatus	Status/mode of the command sequence. - 0 Not used - 1 Edit mode - 2 Not used - 3 Execution mode
RecipeType	Type of command sequence. Pfc (prescribed value) Default setting that cannot be changed.
ApprovalTime	Time stamp for approved command sequences. No entry Is not used for the Command Sequencer module.
ApprovalUserName	Name of the user who approved the command sequence. No entry Is not used for the Command Sequencer module.
ApprovalUserID	ID of the user who approved the command sequence. No entry

	Is not used for the Command Sequencer module.
OutdatedTime	Obsolete time for the command sequence. `No entry` Is not used for the Command Sequencer module.
OutdatedUserName	Name of the user who set the command sequence to obsolete. `No entry` Is not used for the Command Sequencer module.
OutdatedUserID	ID of the user who set the command sequence to obsolete. `No entry` Is not used for the Command Sequencer module.
Structure	Please note the detailed description for the content of this node.

Structure node

Each configured command sequence is represented in the XML file with the **Structure_[serial number]** node.

Parameters	Description
NODE	Given text: `zenOn(R) embedded object`
LastObjId	Last-used ID of the command sequence.
ColCount	Total number of columns in the command sequence grid.
RowCount	Total number of rows in the command sequence grid.
CenterColOffset	Start coordinates of the cell with the first element of a command sequence configuration in the command sequence grid. `0` (prescribed value)
CenterRowOffset	Start coordinates of the row with the first element of a command sequence configuration in the command sequence grid. `0` (prescribed value)
ChartObject	Please note the detailed description for the content of this node.

ChartObject node

Each individual element of the command sequence grid is represented in the XML file with the **ChartObject_[serial number]** node.

The respective XML elements differ depending on the element.

Parameters	Description
NODE	Given text: `zenOn(R) embedded object`

Type	Type of element.
	▶ 1
	`Start element`
	▶ 2
	`End element`
	▶ 3
	`Step`
	▶ 4
	`Transition`
	▶ 5
	`Begin branch`
	▶ 6
	`End branch`
	▶ 7
	`Begin parallel branch`
	▶ 8
	`End parallel branch`
	▶ 9
	`Switchgear allocation`
	▶ 10
	`Not used for command sequences`
	▶ 11
	`Line`
	▶ 12
	`Jump target`