



**COPADATA**  
do it your way

# zenon guida

## Batch Control

v.7.00





© 2012 Ing. Punzenberger COPA-DATA GmbH

Tutti i diritti riservati.

Tutti i diritti riservati la distribuzione e la copia - indifferentemente dal metodo - può essere consentita esclusivamente dalla dittaCOPA-DATA. I dati tecnici servono solo per la descrizione del prodotto e non rappresentano in alcun modo parti legali. Modifiche - anche sotto aspetti tecnici sono a noi riservate

# Contenuto

<b>1. Benvenuti nell'help COPA-DATA.....</b>	<b>6</b>
<b>2. Batch Control .....</b>	<b>6</b>
<b>3. Introduction .....</b>	<b>7</b>
<b>4. Terminology .....</b>	<b>9</b>
<b>5. Process.....</b>	<b>11</b>
<b>6. Engineering in the Editor.....</b>	<b>12</b>
6.1 Units .....	13
6.1.1 Tool bar unit tree.....	13
6.1.2 Project tree - context menu unit .....	14
6.1.3 Context menu units .....	14
6.1.4 Context menu unit X.....	15
6.1.5 Detail view unit.....	15
6.1.6 Detail view unit X.....	16
6.2 Phases .....	17
6.2.1 Context menu phase .....	19
6.2.2 Detail view phase .....	19
6.3 Transitions.....	20
6.4 Conditions .....	21
6.4.1 Waiting periods .....	22
6.5 Reactions.....	23
6.5.1 Context menu reactions unit tree .....	24
6.5.2 Detail view reactions .....	24
6.5.3 Events .....	25
6.5.4 Reaction types .....	28
6.6 Tag.....	30
6.6.1 Context menu tag unit tree .....	30
6.6.2 Detail view tag .....	30
6.6.3 Command tag .....	31
6.6.4 Return tags .....	32

6.6.5	Example for status tag .....	32
6.6.6	Execution period.....	35
6.7	Input interlocking.....	36
6.8	Create screen of type Batch Control .....	37
6.9	Screen switch Batch Control .....	45
6.9.1	Column settings.....	45
6.9.2	Prefiltering.....	50
<b>7.</b>	<b>Configure and control in the Runtime.....</b>	<b>51</b>
7.1	Commands and actions.....	52
7.2	Engineering rules for recipes .....	55
7.3	Status Line .....	58
7.4	Recipe types and recipe states.....	58
7.5	Master recipes .....	60
7.5.1	Create master recipe .....	61
7.5.2	PFC recipe .....	63
7.5.3	Matrix recipe .....	100
7.5.4	Master recipe - test mode .....	106
7.5.5	Release master recipe .....	108
7.6	Control recipe .....	108
7.6.1	Tool bar control recipe. ....	108
7.6.2	Create control recipe.....	110
7.6.3	Execute control recipe.....	111
7.7	Manage recipes.....	113
7.7.1	Manage master recipes .....	114
7.7.2	Manage control recipes.....	119
7.7.3	Saving on the hard disk and backup scenarios .....	122
7.8	Validate recipe .....	123
7.9	Recipe Execution Engine (REE).....	124
7.9.1	Symbols and colors in the Runtime .....	124
7.9.2	The REE states .....	125
7.9.3	Mode and mode change.....	127
7.9.4	Step-by-step execution of a recipe and jumps in the recipe .....	128
7.9.5	Process of a phase in detail .....	130
7.9.6	Exit and restart Runtime.....	135
7.9.7	Restart phase.....	136

<b>8. Behavior in the network .....</b>	<b>138</b>
<b>9. Formula editor .....</b>	<b>139</b>
9.1 List status bits .....	145
9.2 Logical operators.....	148
9.3 Bit formula .....	149
9.3.1 Example bit by bit OR function.....	149
9.4 Comparison operators .....	150
9.5 Examples for formulas .....	151
<b>10. Troubleshooting .....</b>	<b>152</b>

# 1. Benvenuti nell'help COPA-DATA

## GUIDA GENERALE

Nel caso in cui non abbiate trovato delle informazioni che cercavate o se avete dei consigli relativi al completamento di questo capitolo dell'help, mandate una Mail a [documentation@copadata.com](mailto:documentation@copadata.com) (<mailto:documentation@copadata.com>).

## SUPPORTO ALLA PROGETTAZIONE

Se avete delle domande concernenti progetti concreti, potete rivolgervi per E-Mail al [support@copadata.com](mailto:support@copadata.com) (<mailto:support@copadata.com>).

## LICENZE E MODULI

Nel caso in cui doveste constatare che avete bisogno di altri moduli o licenze, rivolgetevi ai nostri dipendenti all'indirizzo [sales@copadata.com](mailto:sales@copadata.com) (<mailto:sales@copadata.com>).

# 2. Batch Control

The module Batch Control offers the possibility to automate batch-orientated manufacturing processes for lot products. The module complies to ANSI/ISA-88.01-1995 also know as ANSI/ISA-S88.

For Batch Control there are two different editors available in the Runtime.

- ▶ PFC editor: (auf Seite 64) For complex recipes with branches.
- ▶ Matrix editor: (auf Seite 101) For simple, sequentially or parallel running recipes.

Depending on the license, both or only one editors is available to you.



### Informazioni sulla licenza

*Nell'Editor e nel Runtime (standalone, server, standby e client) è necessaria una licenza.  
Licensing distinguishes after use of:*

- ▶ PFC editor
- ▶ Matrix editor
- ▶ both editors

#### Editor:

*For the engineering in the zenon Editor one of the Batch editors must be licensed.*

#### Runtime:

*For the execution in the Runtime one of the two Batch editors must be licensed in order to fill the list of recipes. Creating and editing recipes is only possible for the licensed editor.*

#### Network:

*In the network the license of the Server counts. Stand-alone licenses of the Clients are ignored in network operation.*



### Attenzione

*Hints in the property help about VBA keywords and XML export are currently without function.*

## 3. Introduction

The module Batch Control consists of three parts:

1. The engineering environment (auf Seite 12) in the zenon Editor:

There all units (ISA nomenclature, chapter 4.2.5: units) with their phases (ISA - 5.1.2.4: phases) and reactions are created. The phases must have an equivalent in the control (ISA: equipment control) which is called process action according to ISA.

Batch Control reflects the physical model in accordance with ISA 4.2 as flat hierarchical level based on units.

The other levels of the model such as process cell, area, plant, etc. were deliberately forgone. At the creation of the Batch recipes only the lowest level (phase) of the ISA structure model 5.1 was implemented. Additional levels such as operations, unit procedures and procedures are not available.

2. PFC editor (auf Seite 64) and Matrix editor (auf Seite 101):

With the help of these editors you can create master recipes in the zenon Runtime. They form the base of the control recipes which can be executed (see also ISA 5.3.1.). During the process the exact status of the Batch recipe is displayed in the respective editor and you can interfere in the recipe process.

3. Recipe Execution Engine (auf Seite 124) (REE):

The REE is directly integrated in the zenon Runtime and executes a Batch recipe automatically in the background. Via commands such as **Start**, **Pause**, **Stop** etc. the user can control the REE.

There are three possible modes: **Automatic**, **Semi-automatic** and **Manual**.

## **SPECIAL FEATURES OF MODULE BATCH CONTROL:**

In contrast to most other zenon modules, a large part of the engineering - the recipe creation - is done in the Runtime and not in the Editor. This entails special features which are dealt with in the respective chapter. So for example changed phases are no longer transferred to an already released master recipe in order to prevent unwanted data changes.

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 drivers with any PLCs or even RTUs. They only execute the process actions. The entire recipe processing is done at the computer in the REE. For changes on the Batch recipe or for new master recipes, no modifications are necessary in the PLC code.

The module strictly follows the division between the processing of the Batch recipe (ISA: Procedural Control Model) and die execution of the process actions (ISA: Process Model) as described in ISA-S88 chapter 5.2.1.



## PRINCIPLE STRUCTURE OF THE COMMUNICATION

For the communication with the process standard zenon variables are used. As variable names often have cryptic label, an additional abstraction level was introduced. It contains the tags which are available in two types:

- ▶ **Command tag:** They are used to transfer set values to the PLC when a phase is executed.
- ▶ **Return tags:** They are used to return values from the PLC for evaluation.

With both tag types values can be both written (e.g. in reactions) and also read (e.g. in conditions).

The PLC communicates with the zenon driver. The driver communicates with the zenon Runtime. The Runtime sends the values to the REE where they are processed. The REE internally works asynchronously to the zenon Runtime in a 100 ms cycle.

## NETWORK

The module **Batch Control** is fully capable of using a network in terms of Client/Server technology. This means that Batch recipes can be created, duplicated, edited, deleted, etc. on a Client. The whole recipe management remains always on the server. Likewise the whole process control such as **start** recipe, **pause** recipe, **stop** recipe, etc. can be done from the Client. Also mode changes and manual operations such as **jump** are possible.

### **Attenzione**

*Module Batch Control does not support redundancy. There is no synchronization between Standby-Server. When the Server breaks down, the executed Batch recipes are not continued seamlessly on the Standby!*

## 4. Terminology

In the zenon module Batch Control the following terms are used:

Term	Definition
Unit	Physically available machine or equipment part with which phases can be carried out.
Releasing the unit	Element of module Batch Control which cancels the allocation of a unit in the unit manager. With this the unit can be allocated by another recipe again.
Unit allocation	Element of module Batch Control which causes the allocation of a unit in the unit manager. An allocated unit can only be used by phases with the recipe. With this the unit is locked for phases of other recipes which are executed parallel.
Begin parallel branch	Element which ensures the separation of the recipe process in two or more branches.
Begin branch	Element which makes it possible to separate a recipe in two or more branches of which only one can be active at a time. Each following branch must start with a transition. The transition defines which branch is executed in the recipe process.
Begin element	Element of module Batch Control with which every recipe begins.
Batch Control	Tool for creating master recipes and creating and executing control recipes in accordance with ISA-88.
Batch processing	Automatic and sequential processing of a stack of single operations.
End parallel branch	Element which brings together the branches to one branch.
End branch	Element which brings together a branch started by a begin branch element.
End element	Element of module Batch Control with which every recipe ends.
Phase	Predefined process consisting of input interlocking, command and return tags, a phase done condition, event reactions, etc...
Parallel branch	Area of module Batch Control. A parallel branch starts with a begin parallel branch element and is brought together with an end parallel branch element to one execution branch. Between there are at least two branches which are executed parallel.
REE - Recipe Execution Engine	Part of module Batch Control for process control of recipes. The engine executes a control recipe and manages the entire process of the recipe.
Control recipe	Part of module zenon Batch Control. Contains the process of a production process on basis of the batch process according to standard ISA S88. A control recipe is always absed on a master recipe and can be executed once.

<b>Transition</b>	Element of module Batch Control which contains a condition. The element is used after phases in order to ensure a defined transition from one phase to another.
<b>Connection piece</b>	Consists of connection line and connection point. Enables the docking at <b>Begin/end branch/parallel branch</b> .
<b>Connection point</b>	Point on which lines can dock in order to connect with each other.
<b>Connection</b>	Area of module Batch Control which ensure a separation in two or more branches of which only one can be active at the recipe process. It is an either/or branch. A branch always starts with a begin branch element and ends with an end branch element.
<b>Master recipe</b>	Part of module zenon Batch Control. Contains the process of a production process on basis of the batch process according to standard ISA S88. A recipe consists of the following components: Phases, transitions, parallel branch, etc...Master recipes are used as templates for control recipes.

## 5. Process

The engineering and the use of module Batch Control takes place in three main steps:

1. Engineering in the zenon Editor
2. Recipe creation in the Batch editor (PFC or matrix) in the Runtime
3. Recipe execution in the Runtime

### ENGINEERING

The engineer depicts the existing physical world in the Editor. He defines and creates units (auf Seite 13) and assigns phases (auf Seite 17) with tags (auf Seite 30) to them.

### RECIPE CREATION

In the Runtime the recipe creator defines the master recipes on base of the presets from the Editor. They define the process. At this only the units, phases and tags defined in the engineering environment can be used.

**Exception:** If you activate property `Modificabile in ricetta` in group `Settare valore`

`impostabile` in the Editor, you can modify the corresponding tag. This is only possible for command tags.

## RECIPE EXECUTION

The operator executes the recipe in the Runtime. For this he starts a control recipe which is based on the control recipe. Each control recipe can only be started and used once. With this it can be assigned to a lot unambiguously. Recipe either run automatically, semi-automatically or manually.

The operator cannot influence the recipes.

**Exception:** If the recipe creator activates option (auf Seite 79) `Changeable in the control recipe`, the operator can edit tags in the Runtime.

## 6. Engineering in the Editor

To use module Batch Control in the Runtime, you must do the following in the Editor:

- ▶ create units (auf Seite 13), phases (auf Seite 17) and reactions (auf Seite 23)
- ▶ create a screen (auf Seite 37) of type `Batch Control`
- ▶ create a screen switch function (auf Seite 45) for the screen of type `Batch Control`

## DETAIL VIEW

In module Batch Control the detail view divided in two:

- ▶ The left part features the unit tree. The entry `unit` is the starting point of the tree. On the next level the existing units are displayed. The phase belonging to each unit follow. Each phase has the subitem reactions.
- ▶ In the right part a flat list of the units, phases, tags or reactions is displayed depending on what level is chosen on the left side.

## 6.1 Units

To create a new unit:

1. in the project manager go to node `Batch Control`
2. in the detail view select `Unit`
3. in the context menu select menu item **New unit**
4. a new unit is created in the detail view

### 6.1.1 Tool bar unit tree

The tool bar corresponds to the complete unit tree. Depending in the selected element symbols are available or deactivated.



Tag	Description
<b>Rename</b>	Opens the name field of the selected element for renaming.  Not available for main node <b>Units</b> .
<b>Delete</b>	Deletes selected element after confirmation message.  Not available for main node <b>Units</b> .
<b>Expand all</b>	Displays the entire tree structure.  By clicking on the arrow you receive a drop-down list in which you can select one of the following commands: <ul style="list-style-type: none"> <li>▶ Expand all: expands all nodes</li> <li>▶ Collapse all: collapses all nodes</li> <li>▶ Expand selected: expands all selected nodes</li> <li>▶ Collapse selected: collapses all selected nodes</li> </ul> A click on the button always expands all elements.  Via double click on the superordinate entry elements can also be expanded or collapsed.
<b>Help</b>	Opens online help.

### 6.1.2 Project tree - context menu unit

Tag	Description
<b>New unit</b>	Creates a new unit.
<b>Editor profile</b>	Opens the drop-down list with predefined editor profiles.
<b>Help</b>	Opens online help.

### 6.1.3 Context menu units

Right click the main entry **unit** in the unit tree to open a context menu:

Menu item	Action
New unit	Creates a new unit.
Help	Opens online help.

#### 6.1.4 Context menu unit X

Right click on a created unit in order to open the context menu:

Menu item	Action
New phase	Creates a new phase.
Rename	Makes it possible to rename the selected unit.
Delete	Deletes selected unit from the tree after a confirmation message.
Help	Opens online help.

#### 6.1.5 Detail view unit



Menu item	Action
<b>New phase</b>	Creates a new phase in the detail view.
<b>Copy</b>	Copies the selected entries to the clipboard.
<b>Paste</b>	Pastes the content from the clipboard. If an entry with the same name already exists, the content is pasted as "Copy of".
<b>Delete</b>	Deletes selected entries.
<b>Remove all filters</b>	Removes all filter settings.
<b>Edit selected cell</b>	Opens the selected cell for editing. You can see which cell is selected in a line with the help of the binocular symbol in the header.
<b>Replace text in selected column</b>	Opens the dialog for searching and replacing.
<b>Properties</b>	Opens the property window.
<b>Help</b>	Opens online help.

### 6.1.6 Detail view unit X





Tag	Description
New phase	Creates a new phase in the detail view.
Copy	Copies the selected entries to the clipboard.
Paste	Pastes the content from the clipboard. If an entry with the same name already exists, the content is pasted as "Copy of".
Delete	Deletes selected entries.
Remove all filters	Removes all filter settings.
Edit selected cell	Opens the selected cell for editing. You can see which cell is selected in a line with the help of the binocular symbol in the header.
Replace text in selected line	Opens the dialog for searching and replacing.
Properties	Opens the property window.
Help	Opens online help.

## 6.2 Phases

The phase is the execution object of a recipe and therefore the main component. Each phase in module Batch Control must stand facing a **process action** in the control.

**Example:** You want to heat up a tank. For this you need:

- ▶ Firstly the corresponding equipment: a heating in the tank.
- ▶ Then a temperature sensor which measures the actual temperature in the tank. Connect this sensor with a control.
- ▶ In the control a program which controls the heating until the set temperature is reached. This control program is the **process action** in the PLC. It:
  - has an input tag - a set temperature - which is implemented via a command tag (auf Seite 31) in Batch Control
  - needs an output tag - also a set temperature - which must be reached and which is implemented via a return tag (auf Seite 32) in Batch Control

To inform the control about the progress of the recipe, you need corresponding status information which is transferred to the control. For this you use reactions (auf Seite 23) and conditions (auf Seite 21) for the response.

A phase therefore consists of:

Type	Description
Command tags (auf Seite 31)	The set values which should be transferred to the control
Return tags (auf Seite 32)	The response values which inform the REE about the status of the <b>process action</b> in the control. They can be evaluated in conditions and transitions.
Reactions (auf Seite 23)	REE events can be used on the one hand to inform the <b>process action</b> in the control about the state of the REE and on the other hand to inform the user about errors (e.g. time outs, invalid tag values).
Conditions (auf Seite 21)	Are used for the evaluation of the return tags: The state of the <b>process action</b> in the control is evaluated.
Times (auf Seite 22)	Time critical processes can be monitored with this. If the engineered time is exceeded, an event is triggered on which you can react with a reaction.

## ENGINEERING

To create a new phase:

1. select the unit for the phase or first create the desired unit
2. in the context menu select menu item **New phase**
3. a new phase is shown in the detail view
4. the subitem **Reactions** is automatically added to the phase



### Informazioni

*If several phases are selected in the Editor at once whose formulas are identical but whose tags are different, this is not displayed by color coding the different values.*

### 6.2.1 Context menu phase

Right click on a created unit in order to open the context menu:

Menu item	Action
New phase	Creates a new phase.
Rename	Makes it possible to rename the selected unit.
Delete	Deletes selected unit from the tree after a confirmation message.
Help	Opens online help.

#### CONTEXT MENU PHASE X

Right click on a created phase in order to open a context menu for creating the tags:

Tag	Description
New command tag	Creates a new command tag (auf Seite 31).
New return tag	Creates a new return tag (auf Seite 32).
Rename	Makes it possible to change the name of the currently selected phase.
Delete	Deletes the currently selected phase after a confirmation message.
Help	Opens online help.

### 6.2.2 Detail view phase



Tag	Description
New command tag	Creates a new command tag in the detail view.
New return tag	Creates a new return tag in the detail view.
Copy	Copies the selected entries to the clipboard.
Paste	Pastes the content from the clipboard. If an entry with the same name already exists, the content is pasted as "Copy of".
Delete	Deletes selected entries.
Remove all filters	Removes all filter settings.
Edit selected cell	Opens the selected cell for editing. You can see which cell is selected in a line with the help of the binocular symbol in the header.
Edit text in selected line	Opens the dialog for searching and replacing.
Properties	Opens the property window.
Help	Opens online help.

## 6.3 Transitions

Transitions are used in order to ensure a defined transition from one phase to another. The element contains a condition (auf Seite 21) as element and is used as after a phase.

As transitions the following properties are available

- ▶ In pausa :
  - Within the phase the process stops at:
    - Waiting for Finished
    - Waiting for allocation
    - Waiting for interlocking
    - Waiting for Phase done condition
    - Check for parallel execution
- ▶ Fermato
  - Within the phase the process stops at:
    - Waiting for Finished

- Waiting for allocation
  - Waiting for interlocking
  - Waiting for `Phase done condition`
  - Check for parallel execution
- ▶ Stoppato
  - ▶ Annullato
  - ▶ Riavviato
- Phase is completely restarted.

## VARIABLES

Transition conditions can be linked with a binary variable which defines when the phase changes its status. The status changes is delayed until:

- ▶ the value of the variable is defined
- ▶ the value of the variable is 1
- ▶ the status of the variable is not invalid.

If no variable is defined, the status change is always allowed.

All variables for the status change are requested at the advising of the variables in order to receive a value as soon as possible. The values for a variable which define a status change are read on the change to the transient status. A possible pulse must have value `TRUE` within the waiting period in order to be recognized.

**Note:** When closing the Runtime, it is not waited for the variable for the status change from `stopping` to `stopped` as at this time all variables are already signed off.

## 6.4 Conditions

The conditions are used to inform the REE about the status of the `process action` in the control. For evaluating the conditions formulas are used which were created with the formula editor (auf Seite 139).

**Hint:** Use a single status tag (return tag) which takes on different values in order to transmit the status of the `process action` in the control to the phase. You can find an example in chapter Example for status parameter (auf Seite 32).

## ENGINEERING

To create a new reaction:

1. click on the corresponding phase
2. in the properties select the desired conditions from node `Generale` or `Condizione stato transizione`
3. click in the field for the value or on button ...
4. the formula editor is started
5. define the formula (auf Seite 139) for the condition

**Note:** The counterpart to the conditions are the reactions (auf Seite 23). With them the status of the REE is transmitted to the process actions in the control.

### 6.4.1 Waiting periods

The recipe creator can define waiting periods. The configuration of waiting periods (time outs) prevents that time-critical processes take too long because of unforeseen events. If the condition is not fulfilled within the defined waiting period, a corresponding event (auf Seite 25) is triggered. With the reactions (auf Seite 28) you can react on the event and influence the recipe process.

For all waiting periods the following is true:

- ▶ If `0d 00:00:00` is defined as waiting period, the event is not triggered.
- ▶ The waiting periods are independent of the recipe status (e.g. `Recipe paused`) and continue to run even when the Runtime is closed and restarted.
- ▶ If a phase is held and restarted, the waiting periods are also restarted.
- ▶ If a phase is passed through several times, the waiting periods are started again for every pass.
- ▶ Waiting periods themselves do not influence the process. They are simply used to generate an event. The reaction must be defined in the event. After the event is triggered, it is still waited for the fulfillment of the condition.

## 6.5 Reactions

Reactions are the most important piece to influence the recipe process and to communicate with the control. Reactions are always based on events. This can be events of the REE (e.g.: `Phase started`) and also general events (e.g. `Exit Runtime initiated`). With the help of reactions you can e.g. tell the control when a phase has been started or finished in the REE and when all command tags have been written.

Likewise you must transfer the status of the phase to the control with the help of the reactions. Otherwise the control has no information about the process of the recipe.

**Example:** If you stop the REE or the phase, the event `Status change: stop` will be triggered. As reaction you can transfer this status change as set value input to the control. Only then can the control react and stop the `process action`. You can find an example in chapter Example for status parameter (auf Seite 32).

**Note:** The counterpart to the reactions are the conditions (auf Seite 21). With them the states of the process action in the control is transferred to the REE.

### ENGINEERING

With each phase the node **Reactions** is created automatically. In this node you can create any reactions. To create a new reaction:

1. click on **reactions**
2. in the context menu select menu item **New reaction**
3. a new reaction is shown in the detail view
4. in the detail view click on the entry in column **Event**
5. select the desired event from the drop-down list and define the desired reaction type (auf Seite 28) in the property window.

For each reaction type several reactions are possible. They are sorted at the triggering and are executed in accordance to their priority. At this 1 is the highest priority. Some of the reactions are triggered only once in the process - e.g. time outs. If the phase is restarted, these reactions are also retriggered if necessary.

Reactions can only use tags of their own phase. If reactions are copied from other phases, they try to use tags with the same name of the name phase.

Values of the reactions are logged in the CEL.

### 6.5.1 Context menu reactions unit tree

Tag	Description
New reaction	Creates a new reaction in the detail view.
Help	Opens online help.

### 6.5.2 Detail view reactions





Tag	Description
New reaction	Creates a new reaction in the detail view.
Execution order: Earlier	Moves the reaction forward in the execution order.
Execution order: Later	Moves the reaction backward in the execution order.
Execution order: Change places	Only active if exactly two reactions are chosen. The two selected reactions change their places in the execution order.
Copy	Copies the selected entries to the clipboard.
Paste	Pastes the content from the clipboard. If an entry with the same name already exists, the content is pasted as "Copy of".
Delete	Deletes selected entries.
Remove all filters	Removes all filter settings.
Edit selected cell	Opens the selected cell for editing. You can see which cell is selected in a line with the help of the binocular symbol in the header.
Edit text in selected line	Opens the dialog for searching and replacing.
Properties	Opens the property window.
Help	Opens online help.

### 6.5.3 Events

Each `reaction` is a reaction to an event. The event is defined in property `Event`. From the drop-down list you can select the following events:

Event	Description
Unit allocation not possible	Is triggered if the unit was not allocated successfully at first try.
Finished writing command tags	Is triggered if all command tags have been written. It cannot be guaranteed however that really all tags arrived at the control. It depends on the communication and the respective driver. It can however be assumed.  <b>Recommendation:</b> Use this event to tell the PLC that

	the phase has written all command tags and the PLC can start processing the process actions.
Command tag without value	Is triggered if the command tag should be toggled and the variable linked to the tag does not have a valid initial value.
Input interlocking blocked	Is triggered if the input interlocking was blocked at the first check.  Makes only sense if property <code>Interblocco di ingresso</code> was configured.
Phase completed	Is triggered if the phase was started and now is finished.
Phase finished	Is triggered if the phase is finished. This event is the last reaction of the phase and independent of the reason of the finishing. With this the phase done condition is fulfilled.  This event is also triggered at a restart.
Phase started	Is the first event which is triggered. It is also triggered at a restart.  With this event you tell the PLC that the phase has been started in the REE and that it is likely that the command tags will be written soon.
Phase started multiple times	A phase can only be active once. If it is activated several times in parallel, this event is triggered.
REE mode change: Automatic	The REE switched to mode <code>Automatic</code> .
REE mode change: Manual	The REE switched to mode <code>Manual</code> .
REE mode change: Semi-automatic	The REE switched to mode <code>Semi-automatic</code> .
Exit Runtime initiated	Is triggered if the Runtime is exited. This is an especially critical state for module Batch Control as the recipe process does not stop in the control immediately. Therefore exiting the Runtime is prevented as long as module Batch Control saved all data. A process image is created which can later be used as starting point.  Likewise it is made sure that the tags of action <code>Write set value</code> safely arrive at the control. Internally the phase is paused only when the writing

	<p>confirmation from the driver ensued.</p> <p>For more details about existing the Runtime see chapter: Exit and restart Runtime (auf Seite 135)</p> <p>At this event no reaction types of group Influencing the recipe are possible.</p>
Runtime restart	The Runtime was restarted.
Status change	When the phase changes its status, the corresponding reaction is activated.
Status change: Aborting	The phase is aborted at the moment.
Status change: Aborted	The phase was aborted.
Status change: Finished	The phase is finished.
Status change: Stopped	The phase was stopped.
Status change: Holding	The phase is held at the moment.
Status change: Resuming	<p>The phase is resuming after a break.</p> <p>A status change in the object from Paused to In execution triggers the events Resuming and In execution.</p>
Status change: Stopped	The phase was stopped.
Status change: In execution	The phase is executed.
Status change: Restarting	The phase is restarting at the moment.
Status change: Pausing	The phase is switched to Paused at the moment.
Status change: Paused	The phase is paused.
Status change: Stopping	The phase is stopping at the moment.
Linked variable invalid	<ul style="list-style-type: none"> <li>▶ If the value of a variable with status INVALID should be used, this event is created once per invalid variable and phase.</li> <li>▶ If the variable status changed from Invalid to not INVALID and back to INVALID, the reaction is again triggered when the variable is used.</li> <li>▶ If the phase is restarted, the event is triggered again</li> </ul>

	<p>when an invalid variable is used.</p> <p>At the following activities it is checked for invalid variable:</p> <ul style="list-style-type: none"> <li>▶ Source variable in reaction</li> <li>▶ Variable for phase done condition</li> <li>▶ Variable for input interlocking</li> <li>▶ Write command tag inversely</li> <li>▶ Variables for status change at transient states allowed</li> </ul>
Waiting period unit allocation exceeded	Is triggered if the waiting duration for the unit allocation runs. Can also occur during Paused and Held.
Waiting period input interlocking exceeded	<p>Is triggered if the waiting duration (time out) for the input interlocking expired.</p> <p>Makes only sense if a <code>Interblocco di ingresso</code> was defined.</p>
Waiting period following condition exceeded	Is triggered if the phase was not finished within the scheduled waiting duration (time out) although the phase done condition was fulfilled.
Waiting period phase-done condition exceeded	Is triggered if the waiting duration (time out) for waiting for the phase done condition ( <code>Finished</code> ) was exceeded.

#### 6.5.4 Reaction types

In the properties of the reactions the reaction types more precisely defined and engineered. In group `Sostituzioni` the following reaction types are available:

Reaction type	Description
Impostazione parametri di valore	<p>Influences command and return tag directly. All tag data types can be used.</p> <p><b>Attention:</b> The value must be within of the set value limits of the variable which is linked at the tag. If this is not the case, an error message is created during the validation.</p>
Inserimenti nella CEL	<p>Creates entries in the CEL and log files. With this the reaction can be documented and the recipe process can be tracked later. For this property <code>Crea inserimento CEL</code> must be activated. The text for the CEL is defined in property <code>Messaggio di testo CEL</code>.</p>
Funzione	<p>Makes it possible to link any zenon function.</p> <p>With this you can e.g. call up a pop-up in order to inform the user about a certain status or start a data backup.</p> <p><b>Note:</b> In the network the function is always executed at the server.</p>
Assegna parametro	<p>Makes it possible to perform a value assignment from <code>Parametro sorgente</code> to another <code>Parametro di destinazione</code>. You can use both command tags and return tags. The data type of source and target tag must be identical otherwise an error is displayed at the validation of the recipe.</p>
Influsso ricetta	<p>Make it possible to:</p> <ul style="list-style-type: none"> <li>▶ change the REE mode</li> <li>▶ execute REE commands</li> <li>▶ execute phase commands</li> </ul> <p>With this you can react on serious events such as <code>Waiting duration exceeded</code> or <code>Linked variable invalid</code>.</p> <p><b>Note:</b> Use this reaction type carefully as this reaction type influence the entire recipe process.</p> <p>For each event you can only once:</p> <ul style="list-style-type: none"> <li>▶ set the <code>Modalità REE</code> and</li> <li>▶ write a single <code>Comando</code></li> </ul> <p>Because e.g. it does not make sense to pause and hold the recipe at the same time with the same event.</p>

## 6.6 Tag

Tags are the communication interface to the control. With them all values are transferred to the control and also read back. To not have to work with complex and for the user incomprehensible variable names in module Batch Control, the abstract level is used. Each tag consists - for each phase - of a unique name and a description. With this the engineer can give the recipe creator or operator a description for what the tag is used or which effects it has.

You can add any number of tags to a phase. We distinguish between command tags (auf Seite 31) and return tags (auf Seite 32). Each tag can be switched between command and return tag at any time.

### 6.6.1 Context menu tag unit tree

Right click on a created phase in order to open a context menu for creating the tags:

Tag	Description
New command tag	Creates a new command tag (auf Seite 31).
New return tag	Creates a new return tag (auf Seite 32).
Rename	Makes it possible to change the name of the currently selected phase.
Delete	Deletes the currently selected phase after a confirmation message.
Help	Opens online help.

### 6.6.2 Detail view tag



Tag	Description
New command tag	Creates a new command tag in the detail view.
New return tag	Creates a new return tag in the detail view.
Copy	Copies the selected entries to the clipboard.
Paste	Pastes the content from the clipboard. If an entry with the same name already exists, the content is pasted as "Copy of".
Delete	Deletes selected entries.
Remove all filters	Removes all filter settings.
Edit selected cell	Opens the selected cell for editing. You can see which cell is selected in a line with the help of the binocular symbol in the header.
Edit text in selected line	Opens the dialog for searching and replacing.
Properties	Opens the property window.
Help	Opens online help.

### 6.6.3 Command tag

Command tags contain the set values which should be transferred to the control. At the execution of the phase all command tags are written to the control. You can find the exact description in chapter: Process of a phase in detail (auf Seite 130).

Command tags can also be used in transitions (auf Seite 20), conditions (auf Seite 21) and reactions (auf Seite 23).

Command tags have a number of properties which can be defined via the property window. Each tag must be linked with a variable. The data type of the variable must comply with the data type of the tag and the set value limits of the parameters must be within the set value limits of the variable. If this is not the case, error messages are created during the validation.

**Hint to property `Modificabile in ricetta`:** With this you define whether the value of the command tag may be modified by the creator of the master recipe. If e.g. machine tags should not be changeable in the recipe but defined fixedly, you must deactivate this property.

## ENGINEERING

To create a new command tag:

1. select the desired phase
2. in the context menu select menu item **New command tag**
3. a new command tag is created in the detail view

### 6.6.4 Return tags

The return tags contain the return values with which the process action of the REE communicate its status. Normally the value is set by the control and evaluated by the REE. Return tags can be evaluated in transitions (auf Seite 20) and conditions (auf Seite 21).

Likewise they can be used in reactions (auf Seite 23) and can also be written there. For this they are listed as **target tags** at **Assegna parametro** and **Impostazione parametri di valore**.

## ENGINEERING

To create a new return tag:

1. select the desired phase for which you want to create a new return tag
2. in the context menu select menu item **New return tag**
3. a new command tag is created in the detail view

### 6.6.5 Example for status tag

To be able to communicate with the control, you normally need two status tags:

- ▶ one in write direction and
- ▶ one in read direction



The variable behind these parameters should have a numeric data type such as USINT or UINT.

We recommend that you execute both tags as return tags. This may seem illogic for commands in write direction at first glance but has the following background: At the execution of the phase, all command tags are written. With this they are visible in the list of command tags and can therefore be deleted accidentally. This makes no sense for a command for the control. The goal is not just to communicate a single value to the control as command but to transmit the status of the phase in the the recipe.

Especially at the writing of a command tag it makes sense to not simply inform the PLC about the writing but the status when all values have been written and the PLC therefore can start to process the process action.

For this it is best to use the reaction to event `Finished writing command tags`. At reactions to an event you can also write values to an return tag. Therefore it is better to use return tags for both status tags.

Here is an example about which values the tags can take on:

**STATUS TAG IN WRITE DIRECTION (TO THE PLC): COMMANDS**

Value	Name of the event
0	not defined
1	Phase started
2	Finished writing command tags
3	Phase finished: Phase done condition fulfilled and Minima durata di esecuzione reached (if engineered)
4	Phase completed
5–9	Reserve
10	Status change: pausing
11	Status change: Resuming
12	Status change: Holding
13	Status change: Restarting
14	Status change: Stopping
15	Status change: Aborting
16–19	Reserve
20	REE mode change: Automatic
21	REE mode change: Semi-automatic
22	REE mode change: Manual
23–29	Reserve
30	Exit Runtime initiated
31	Runtime restart
32	Unit allocation not possible
33	Waiting period unit allocation exceeded
34	Input interlocking blocked
35	Waiting period input interlocking exceeded
36	Waiting period phase-done condition exceeded

37	Waiting period following condition exceeded
38	Phase started multiple times

You can find the exact meaning of the events in chapter Event type (auf Seite 25).

For each entry in the table you define a corresponding reaction for writing the status value at the phase.

**Hint:** Use the same tag label for all phases; e.g. `StatusPhase`. Then you only have to engineer the reaction at one phase and can then transfer it to all phases via copy & paste. You can of course also copy the tags. Do not forget to correct the variable. They must match the respective phase.

#### TAG IN READ DIRECTION (FROM THE PLC): RETURN VALUES

Value	Description	Linked in property
0 - 1	Not defined	
2	Process actionfinished	Condizione di termine fase
3 - 9	Reserve	
10	Process action paused	In pausa
11	Reserve	
12	Process action held	Fermato
13	Process action restarted	Riavviato
14	Process action stopped	Stoppato
15	Process action aborted	Annulato

Link the values with a formula in the respective property.

**Hint:** You can copy the formula and just change the respective value. If you make this setting at the beginning of the engineering of the first phase, you can copy the entire phase and with that have this settings for all phases.

### 6.6.6 Execution period

The execution duration is controlled via two independent properties. Their values must not complement one another.

## MAXIMUM EXECUTION DURATION

The `Massima durata di esecuzione` refers to Phase completed and therefore to the process. It is not connected to the Minimum execution duration.

## MINIMUM EXECUTION DURATION

Property `Minima durata di esecuzione` defines how long zenon waits after writing the command tag independent of the check of the phase done condition. During the execution the minimum execution duration is checked. If the value is larger than 0, an event is triggered after it has expired. This happens regardless of whether the phase still checks its `Condizione di termine fase` or only waits for the `Minima durata di esecuzione`.

The length of the minimum execution duration can exceed the maximum execution duration.

## EXAMPLE

- ▶ There is a phase: `start mixing`. The confirmation that the mixer runs must not take longer than 5 seconds before a warning of an error is displayed.  
Projecting: Property `Massima durata di esecuzione` gets value 5 seconds with corresponding reaction.
- ▶ The mixer however should run 15 minutes before the next phase is executed.  
Projecting: Property `Minima durata di esecuzione` gets value 15 minutes.

With this the minimum execution duration is 15 minutes and the maximum execution duration 5 seconds.

## 6.7 Input interlocking

With the help of an input interlocking the phase is only executed in the Runtime when the condition for the input interlocking is fulfilled.

The input interlocking is configured via property `Interblocco di ingresso`. Via the formula editor (auf Seite 139) the condition is defined which the input interlocking must fulfill. The formula can consist of one or more command tags and return tags of the phase. Value and status of the variables can also be used. The formula returns `TRUE` or `FALSE` as result. The condition can be displayed in the Runtime but cannot be changed there. The waiting period for the input interlocking is configured with the help of property `Tempo di attesa per interblocco di ingresso`.

If an input interlocking was defined, the phase is executed as soon as the following conditions are fulfilled:

- ▶ the phase is active
- ▶ the unit is allocated
- ▶ the phase is not active twice
- ▶ the phase is not already executed
- ▶ the input interlocking is fulfilled

If no input interlocking is linked, the phase is executed when:

- ▶ the phase becomes active
- ▶ the unit is allocated
- ▶ the phase is not executed twice
- ▶ the needed variables all have a value



#### Informazioni

*You can find more information about input interlockings in chapter Processing a phase in detail (auf Seite 130).*

## 6.8 Create screen of type Batch Control

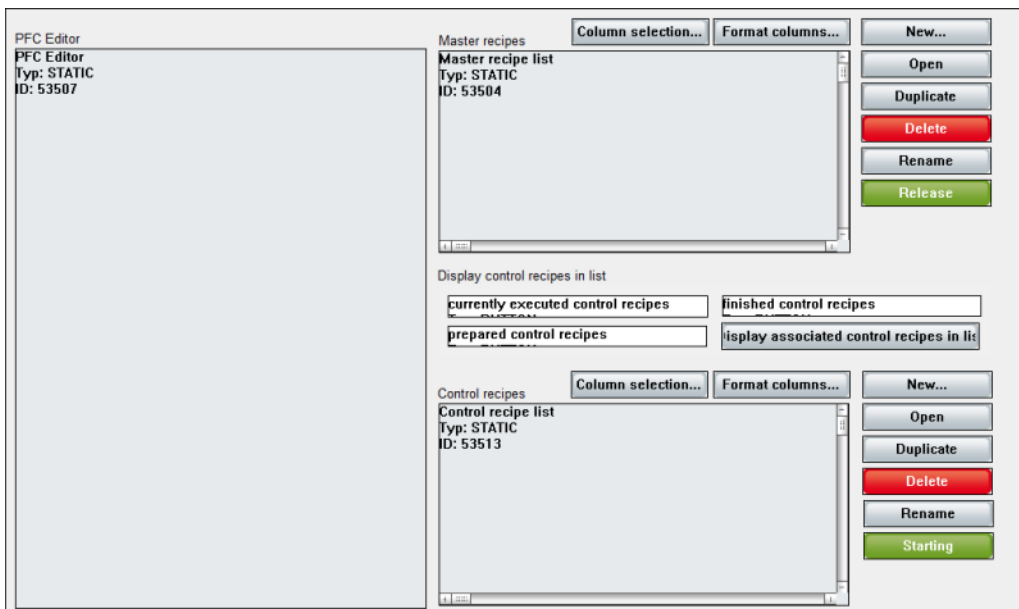
In screen of the type Batch Control master recipes (auf Seite 60) and/or control recipes (auf Seite 108) can be:

- ▶ created
- ▶ managed
- ▶ executed in the Runtime

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

To create a screen of type Batch Control:

1. in node **screens** in the context menu select menu item **New screen**
2. in column **Screen type** select **Batch Control** from the drop-down list
3. in the menu bar **Control elements** select one of the three default settings:
  - Insert template: provides templates for different resolutions and selection of only master recipes or default (master and control recipes)
  - Default master recipe: adds selected elements for a master recipe
  - Default control recipe: adds selected elements for a control recipe
4. the default elements for the desired settings are created
5. if needed you can add additional control elements from menu **Control elements**
6. create a screen switch (auf Seite 45) in order to display the screen in the Runtime



Element	Description
Insert template	Inserts control elements for master recipes and/or control recipes on predefined locations on the screen. These control elements can be supplemented, reduced and positioned newly.
Default (master recipe)	Inserts control elements for master recipes on predefined locations on the screen. These control elements can be supplemented, reduced and positioned newly.
Default (control recipe)	Inserts control elements for control recipes on predefined locations on the screen. These control elements can be supplemented, reduced and positioned newly.
Recipe editor	Adds the licenses editor for creating master and control recipes.
Master recipes	Control elements for master recipes.
List master recipes	<p>In this list all master recipes can be displayed. The display can be limited by filters to an individual selection.</p> <p>The filtering can be preset in the zenon Editor in the screen switch function (auf Seite 45). Online filtering is also possible. These filters are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p> <p>All commands are also possible in the context menu of the list. The commands for list management can be called from the header of the list. The commands for recipe management can be called at editing one or more recipes.</p> <p>The recipes in the list cannot be edited directly in the list. Renaming, changing the description or changing the recipe status is only possible with the corresponding commands.</p>
Column selection master recipe...	<p>Opens a dialog in order to determine which columns should be displayed (auf Seite 47).</p> <p><b>Attention:</b> These changes are discarded when the screen is called up again. A permanent definition is only</p>

	possible in the zenon Editor.
<b>Format columns master recipe...</b>	<p>Opens a dialog to edit the column settings (auf Seite 48).</p> <p><b>Attention:</b> These changes are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p>
<b>New master recipe...</b>	<p>Opens dialog for creating a new master recipe (auf Seite 61).</p>
<b>Rename master recipe</b>	<p>Only active if exactly one master recipe is selected. The dialog for the input of a unique name and the description is opened.</p> <p>Recipes can only be renamed if they are in status <code>Editable</code>.</p> <p>Also use this function in order to changed the description of the master recipe.</p> <p>When renaming a recipe, a CEL entry is created.</p>
<b>Duplicate master recipe</b>	<p>Only active if exactly one recipe is selected. Created a copy of the selected recipe. At the creation of the copy, the version of the recipe saved on the hard disk is used. If the recipe is just edited in another computer and the changes have not yet been saved, the changes are not applied. The dialog for the input of a unique name and the description is opened.</p> <p>The copy of the recipe automatically receives status <code>Editable</code> and can be edited further.</p> <p>When duplicating a recipe, a CEL entry is created.</p>
<b>Delete master recipe</b>	<p>Deletes the selected recipes irrevocably. If the recipe is opened on another computer for editing, it is automatically closed there.</p> <p>Deleting is only possible if there are no control recipes which are based on the master recipe. First you must delete all control recipes.</p> <p>Recipes which are currently executed in test mode (master recipe status: <code>Test in execution</code>) cannot be deleted. First they must be <code>finished</code>, <code>stopped</code> or <code>canceled</code>.</p> <p>If recipes must not be deleted - e.g. FDA regimented environment - we recommend not to engineer this button or to give it an appropriated <code>Livello di autorizzazione</code>.</p>



	When deleting a recipe, a CEL entry is created.
<b>Open master recipe</b>	Opens the selected master recipe in the recipe editor if screen element <code>Recipe editor</code> exists in the screen. Each selected master recipe is opened in a separate tab of the recipe editor.
<b>Switch master recipe to edit mode</b>	Changes the master recipe status of the selected recipes to <code>Editable</code> . In this status recipe can again be edited completely. Only recipes in <code>Test mode</code> can be set back to <code>Editable</code> .
<b>Switch master recipe to test mode</b>	Changes the master recipe status of the selected recipe to <code>Test mode</code> . Only faultless recipes can be switched to test mode. If error occur during the validation (auf Seite 123), you must first fix them.  Recipes in the test mode can be executed but no longer reengineered. For details about the states see chapter Recipe types and recipe states (auf Seite 58).
<b>Release master recipe</b>	Changes the master recipe status of the selected recipes to <code>Released</code> . Only faultless recipes can be released. If error occur during the validation (auf Seite 123), you must first fix them. Only recipes in status <code>Test mode</code> and <code>Editable</code> can be released.  Released recipes can no longer be edited. Control recipes can only be created from released recipes. For details about the states see chapter Recipe types and recipe states (auf Seite 58).  When releasing a recipe, a CEL entry is created.
<b>Display associated control recipes in list</b>	Opens all control recipes which are based on the selected master recipes and which comply with the set filter criteria.
<b>Filter for displaying the control recipe</b>	Makes it possible to filter control recipes for the following criteria: <ul style="list-style-type: none"> <li>▶ <b>Currently executed control recipes:</b> Displays only control recipes which are currently executed. Only takes effect as soon as you click on <b>Show associated control recipes in list</b>.</li> <li>▶ <b>Prepared control recipes:</b></li> </ul>

	<p>Display only control recipes which are prepared for execution. Only takes effect as soon as you click on <b>Show associated control recipes in list</b>.</p> <p>► <b>Completed control recipes:</b> Displays only control recipes which have already been executed. Only takes effect as soon as you click on <b>Show associated control recipes in list</b>.</p>
--	--

Control recipe	Control elements for control recipes.
List control recipes...	<p>In this list all control recipes can be displayed. The display can be limited by filters to an individual selection.</p> <p>Per default the list is empty. For filling the list you must:</p> <ul style="list-style-type: none"> <li>▶ select master recipes</li> <li>▶ set filters  <b>currently executed control recipes,</b>  <b>prepared control recipes</b> and  <b>completed control recipes</b></li> <li>▶ click button <b>display associated control recipes in list</b></li> </ul> <p>In addition to the filters mentioned above, you can filter the list itself. The filtering can be preset in the zenon Editor in the screen switch function (auf Seite 45). Online filtering is also possible. These filters are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p> <p>All commands are also possible in the context menu of the list. The commands for list management can be called from the header of the list. The commands for recipe management can be called at editing one or more recipes.</p> <p>The recipes in the list cannot be edited directly in the list. Renaming, changing the description or starting the recipes is only possible with the corresponding commands.</p>
Column selection control recipe...	<p>Opens a dialog in order to determine which columns should be displayed (auf Seite 47).</p> <p><b>Attention:</b> These changes are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p>
Format columns control recipe...	<p>Opens a dialog to edit the column settings (auf Seite 48).</p> <p><b>Attention:</b> These changes are discarded when the screen is called up again. A permanent definition is only</p>

	possible in the zenon Editor.
<b>New control recipe...</b>	<p>Opens the dialog (auf Seite 110) for entering a unique name and a description for the control recipe. The uniqueness of the name is also checked in the zenon network. The name must only be unique within the master recipes. Control recipes which are based on other master recipes may have the same name. The uniqueness within module Batch Control is achieved by always referencing the master recipe name and the control recipe name.</p> <p>When creating a control recipe, a CEL entry is created.</p>
<b>Rename control recipe</b>	<p>Only active if exactly one control recipe was selected. The dialog for the input of a unique name and the description is opened.</p> <p>Recipes can only be renamed if they are in status <i>Prepared</i>.</p> <p>Also use this function in order to changed the description of the control recipe.</p>
<b>Duplicate control recipe</b>	<p>Only active if exactly one recipe is selected. Created a copy of the selected recipe. At the creation of the copy, the version of the recipe saved on the hard disk is used. If the recipe is just edited in another computer and the changes have not yet been saved, the changes are not applied. The dialog for the input of a unique name and the description is opened.</p> <p>The copy of the recipe automatically gets the status <i>Prepared</i> and can therefore be edited and started. The REE status (auf Seite 125) of the duplicate is set to <i>automatic</i>.</p> <p>When duplicating a recipe, a CEL entry is created.</p>
<b>Delete control recipe</b>	<p>Deletes the selected recipes irrevocably. If the recipe is opened on another computer for editing, it is automatically closed there.</p> <p>Deleting is only possible if all selected recipes are not executed (control recipe status: <i>In execution</i>). First they must be <i>finished</i>, <i>stopped</i> or <i>canceled</i>.</p> <p>If recipes must not be deleted - e.g. FDA regimented environment - we recommend not to engineer this button or to give it an appropriated <i>Livello di</i></p>

	<b>autorizzazione:</b> When deleting a recipe, a CEL entry is created.
<b>Open control recipe</b>	Opens the selected control recipe in the recipe editor if screen element <b>Recipe editor</b> exists in the screen. Each selected control recipe is opened in a separate tab of the recipe editor.
<b>Start control recipe</b>	Starts the selected control recipe in the defined REE mode. The recipes are executed invisibly at the Server. It is not necessary that the recipe is opened in the recipe editor.

## 6.9 Screen switch Batch Control

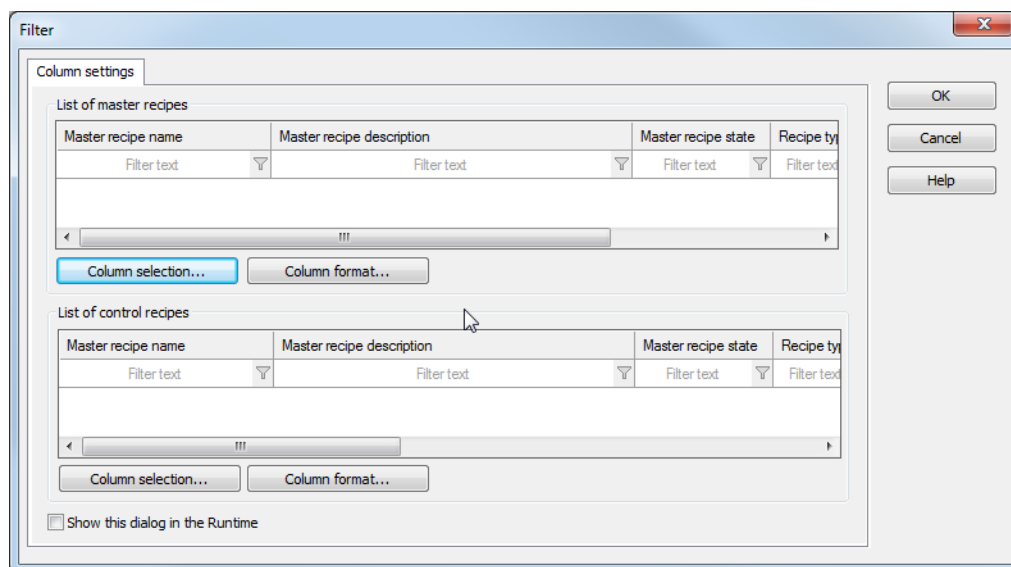
To use Batch Control in the Runtime, engineer a screen switch function to a screen of type **Batch Control**:

1. in node **Functions** in the context menu select menu item **New function**
2. select function **Screen switch**
3. select the screen of type **Batch Control**
4. the filter dialog (auf Seite 45) is displayed
5. configure the column settings (auf Seite 45) for the list of the master recipes/control recipes as you wish
6. configure the desired prefiltering of the master recipes/control recipes
7. link the function with a button on the screen in order to switch in the Runtime

### 6.9.1 Column settings

The column settings are configured for:

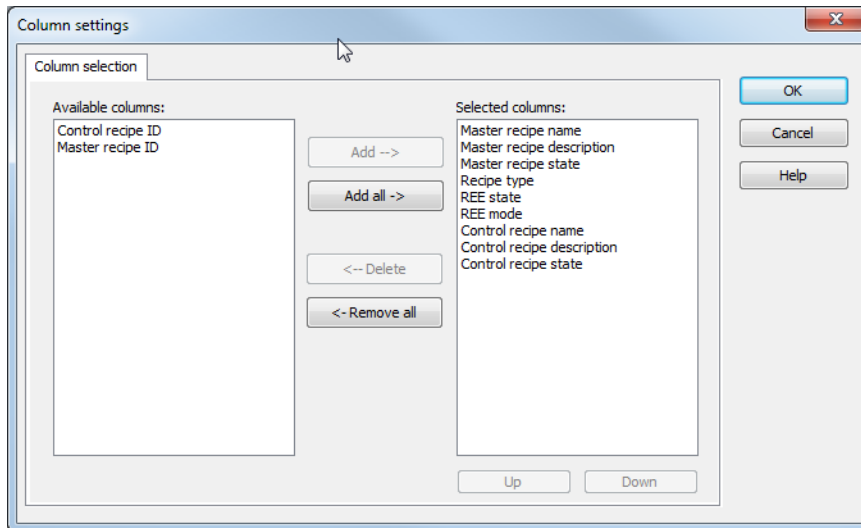
- List of master recipes
- List of control recipes



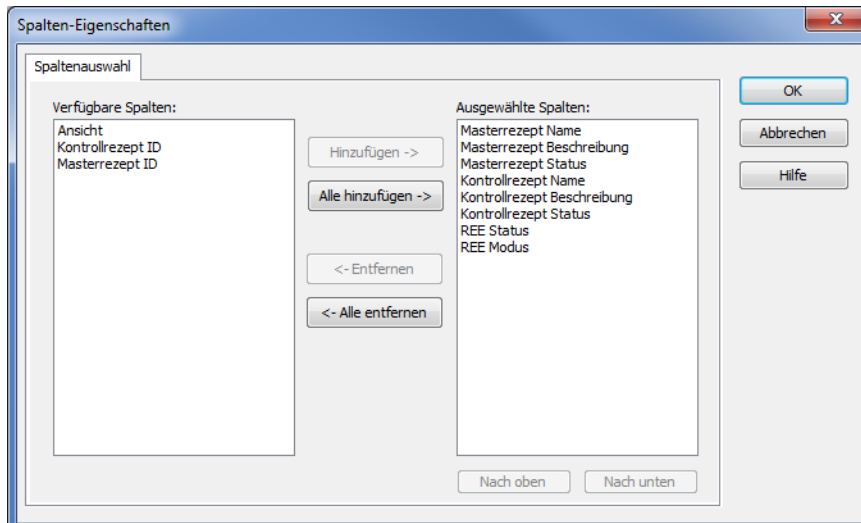
Parametri	Descrizione
<b>Campo lista</b>	Visualizzazione delle colonne configurate.
<b>Selezione colonne</b>	Apri il dialogo per la selezione delle colonne.
<b>Formato colonne</b>	Apri il dialogo che serve a formattare le colonne.
<b>OK</b>	Riprende e accetta tutte le modifiche e chiude il dialogo.
<b>Annulla</b>	Annulla tutte le modifiche e chiude il dialogo.
<b>Guida</b>	Attiva la guida online

## Column selection

Column selection for list of the master recipes:



Column selection for list of the control recipes:



Pulsanti	Funzione
<b>Aggiungi</b>	Sposta le colonne selezionate da quelle disponibili a quelle selezionate. Dopo aver confermato il dialogo cliccando su OK, esse vengono visualizzate nella visualizzazione dettagliata.
<b>Aggiungi tutte</b>	Sposta tutte le colonne disponibili nelle colonne selezionate.
<b>Rimuovi</b>	Elimina le colonne evidenziate dalle selezionate e le visualizza nella lista di quelle disponibili. Dopo aver confermato il dialogo cliccando su OK, esse vengono eliminate dalla visualizzazione dettagliata.
<b>Cancella tutte</b>	Cancella tutti le colonne dalla lista delle colonne scelte.
<b>In sù</b>	Sposta l'inserimento selezionato verso l'alto. Questa funzione è sempre disponibile per inserimenti singoli, non è possibile però nel caso di una selezione multipla.
<b>In giù</b>	Sposta l'inserimento selezionato verso il basso. Questa funzione è sempre disponibile per inserimenti singoli, non è possibile però nel caso di una selezione multipla.
<b>OK</b>	Riprende le impostazioni e chiude il dialogo.
<b>Annulla</b>	Rifiuta le impostazioni e chiude il dialogo.
<b>Guida</b>	Apri la guida online

## Column format

In this dialog you define the column format:





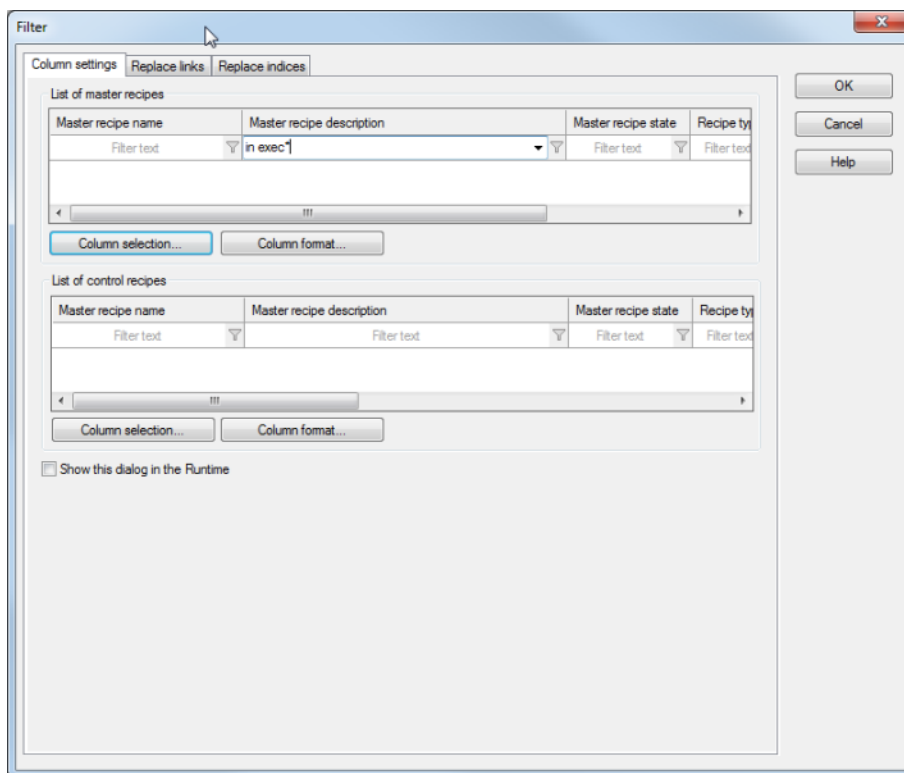
Parametri	Descrizione
Colonne disponibili	Lista delle colonne disponibili facendo uso della <b>Selezione colonne</b> . La colonna selezionata in questa sede viene configurata mediante le proprietà della sezione <b>Parametri</b> .
Parametri	Impostazioni per colonna selezionata.
Descrizione	Nome del titolo della colonna Il titolo colonna è traducibile online. A tal scopo dovete inserire il carattere @ prima del nome.
Larghezza	Larghezza della colonna in caratteri. Calcolo: numero per larghezza media del carattere selezionato.
Orientamento	Orientamento.  Impostazioni possibili: <ul style="list-style-type: none"> <li>▶ Orientamento a sinistra: nella colonna, il testo si dispone prendendo a riferimento il bordo sinistro.</li> <li>▶ Centrato: nella colonna, il testo si dispone in posizione centrale.</li> <li>▶ Orientato a destra: nella colonna, il testo si dispone prendendo a riferimento il bordo destro.</li> </ul>
Blocca il filtro colonna nell'immagine di processo	attivo: Il filtro per questa colonna non può essere modificato durante il Runtime.  <b>Indicazione:</b> disponibile solo per: <ul style="list-style-type: none"> <li>▶ Batch Control</li> <li>▶ Trend esteso</li> <li>▶ Message Control</li> <li>▶ Manager gruppi ricette</li> </ul>
OK	Riprende le impostazioni e chiude il dialogo.
Annulla	Rifiuta le impostazioni e chiude il dialogo.
Guida	Apre la guida online

## 6.9.2 Prefiltering

In order that in the Runtime all recipes are not loaded in the recipe list all the time, you can define filter in the screen switch function. To do this:

1. in the screen switch function click on column filters for the **List of the master recipes** or the **List of control recipes**
2. enter the desired filter text; wildcards (\*) are allowed:  
e.g. for the **List of the master recipes** in column **Master recipe status** the text **In Exec\***
3. confirm the filter text with **Return** for it to be applied.

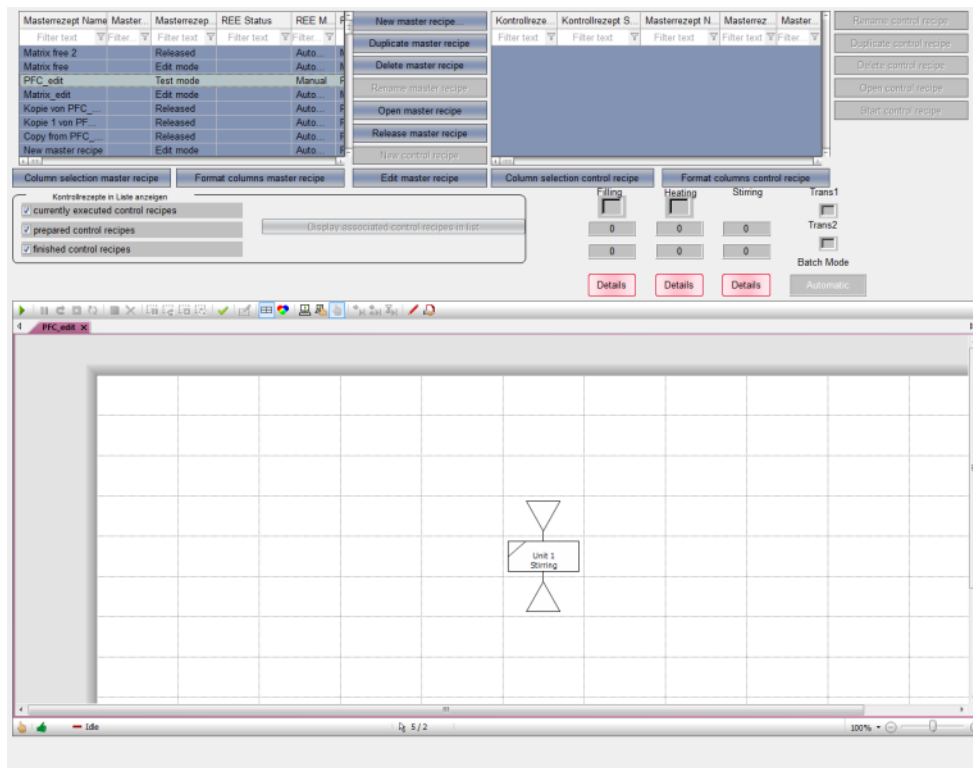
In the Runtime in the list of the master recipes only the recipes are displayed which are currently in test mode and which are currently executed. You can use any number of column filters at a time.



If you want that the user cannot remove the filter in the Runtime, open dialog **column format...** (auf Seite 48) and activate option **Lock column filter** in the process screen. With this the user cannot modify the filters in the Runtime and therefore does not get the recipes, which e.g. he is not allowed to modify, not displayed.

## 7. Configure and control in the Runtime

The entire management, creation and processing of the recipes is done in the Runtime. Editing in the zenon Editor is not possible.



### CHECKING FOR CHANGES IN THE EDITOR

When opening a recipe it is checked whether the engineering of unit, phase, etc. was changed in the Editor. If a change is detected, it is taken over in the object to which the function concerned is assigned. Settings which were overwritten in the Runtime remain the same. When the recipe was changed, it is shown via an asterisk (\*).

At reloading the recipe is also checked.

**Note:** Only recipes in the edit mode are updated.

If a changed recipe should be release which was not yet saved, a prompt is displayed with the following possibilities:

- release the current recipe

- ▶ release the saved recipe
- ▶ cancel the release process



### Attenzione

*Changes to phases in the Editor are taken over without warning message when the recipe is released in the Runtime. For all data which are not overwritten in the Runtime, the Editor is the leading system.*

## SELECTION PROCEDURE IN LISTS:

- ▶ `Ctrl+A`: selects all elements
- ▶ `Ctrl+mouse click`: adds master recipe to the existing selection
- ▶ `Ctrl+Shift+mouse click`: Extends selection from the currently selected master recipe to the clicked master recipe

## 7.1 Commands and actions

In the Runtime the following commands and actions are available:

- ▶ Commands effect the recipe process.
- ▶ Actions make it possible to edit recipes.

### COMMANDS

For a command to be accepted by the phase, the following requirements are necessary:

- ▶ The REE must run.
- ▶ The phases must be active.
- ▶ The phase must be in a state in which the command is allowed.

Via multi-selection the command can be sent to several phases in the same execution cycle.

Command	Description
Start recipe	Starts the recipe process.
Pause recipe	Pauses the recipe process.
Resume recipe	Resumes a held or paused recipe.
Hold recipe	Holds the recipe process.
Restart recipe	Restarts the recipe.
Stop recipe	Stops the recipe.
Abort recipe	Aborts the recipe.
Pause phase	Pauses the phase.
Resume phase	Resumes the phase.
Hold phase	Holds phase.
Restart phase	Restarts the phase.
Switch to automatic mode	Switches the REE to automatic mode.
Switch to semi-automatic mode	Switches the REE to semi-automatic mode.
Switch to manual mode	Switches the REE to manual mode.
Continue recipe only on selected execution position	Continues a recipe at the selected position.
Continue recipe on all execution positions	Continues a recipe on every available position.
Skip active condition	Skips an active condition.  Only possible in the manual mode.

## ACTIONS

Action	Description
Check recipe for errors	Checks the recipe for errors and displays error messages.
Edit element	Opens the corresponding dialog for editing the selected element.
Show/hide grid	Toggles the grid for the PFC editor on/off.
Change background color and grid	Opens the dialog (auf Seite 69) for changing the

<b>color</b>	background color and grid color.
<b>Duplicate Recipe</b>	Copies the selected recipe and adds it as copy to the list.
<b>Create control recipe</b>	Creates a control recipe.
<b>New master recipe</b>	Opens the dialog (auf Seite 61) for creating a new recipe.
<b>Save master recipe</b>	Saves all changes which were done since the last saving.
<b>Delete</b>	Deletes the selected element.
<b>Exchange phase</b>	Opens the dialog (auf Seite 73) for selecting a phase. The present phase is replaced by the newly selected phase.
<b>Edit mode</b>	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 <b>ESC</b> key.
<b>Add phase</b>	Occupies the mouse cursor with a phase (auf Seite 71). It can be added to any allowed, free location via click.
<b>Add transition</b>	Occupies the mouse cursor with a transition (auf Seite 93). It can be added to any allowed, free location via click.
<b>Add begin parallel branch</b>	Occupies the mouse cursor with a begin parallel branch (auf Seite 96). It can be added to any allowed, free location via click.
<b>Add end parallel branch</b>	Occupies the mouse cursor with an end parallel branch (auf Seite 96). It can be added to any allowed, free location via click.
<b>Add begin branch</b>	Occupies the mouse cursor with a begin branch (auf Seite 93). It can be added to any allowed, free location via click.
<b>Add end branch</b>	Occupies the mouse cursor with an end branch (auf Seite 93). It can be added to any allowed, free location via click.
<b>Add unit allocation</b>	Occupies the mouse cursor with a unit allocation (auf Seite 69). It can be added to any allowed, free location via click.
<b>Add jump target</b>	Occupies the mouse cursor with a jump target (auf Seite 99). It can be added to any allowed, free location via click.
<b>Switch recipe to test mode</b>	Switches recipe to the test mode (auf Seite 106).
<b>Release recipe</b>	Releases (auf Seite 108) the recipe. With this a control recipe can be created.

## 7.2 Engineering rules for recipes

At the engineering the rules defined in standard ANSI/ISA-S88 are generally true.

Important principles:

### GENERAL

- ▶ For all elements all connection points must be connected.  
Exception: Jump target element. Here at least the output connection point and one input connection point must be connected.
- ▶ The **begin element** exists exactly once and marks the beginning of the process.
- ▶ The **end element** exists exactly once and marks the end of the process.
- ▶ Phases can be inserted anywhere. You can also place several phases in succession.
- ▶ Two transitions may not lie one after the other.

### BRANCHES

- ▶ The first element after an **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**. Outside of an parallel branch a branch can also end in a jump target element. Any element can be placed between begin and end of an branch even parallel branches as long as they are closed before the **end branch** element.
- ▶ 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**.
- ▶ **End branch** elements can be used in the same way as jump target elements.  
Limitation: The element must not be followed by a transition.

## PARALLEL BRANCH

- ▶ The first element after a `Begin parallel branch` must not be a transition.
- ▶ The individual branches which start at `begin parallel branch` must all end in an `end parallel branch` never 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 can be embedded in parallel branches, For this, the following applies:
  - each embedded parallel branch must unite to the superordinate parallel branch
  - embraced interlacing is not supported

## LINES IN THE PFC EDITOR

- ▶ 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: The both inputs of two phases 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 (auf Seite 123) an error message is displayed.

## JUMP TARGETS

- ▶ Jump targets are used 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 one input and the output must be connected. At this it makes no difference which input connection point is connected.



- ▶ Jump targets may be interlaced consecutively.
- ▶ Jumps are prohibited:
  - between parallel branches
  - to jump out of a parallel branch
  - to jump in a parallel branch

## REACTIONS

- ▶ Reactions can appear in each object state.
- ▶ For each reaction type several reactions are possible. They are sorted at the triggering and are executed in accordance to their priority. At this 1 is the highest priority.
- ▶ Variables which are used in the tags of the reaction as source are registered at the driver as for reading. If a value is needed at a reaction but is not yet available or invalid, the alternate value is written. The writing of the value is done without write confirmation.
- ▶ Some of the reactions are triggered only once in the process - e.g. time outs. If the phase is restarted, these reactions are also retriggered if necessary.
- ▶ If as an reaction to a status change a command was triggered which triggers an immediate status change (command to REE), the command of the second reaction is processed directly after processing the first. Commands processed.

## EXECUTION PERIOD

- ▶ Property `Minima durata di esecuzione` defines how long it is waited after writing the command tag independent of the check of the phase done condition.
- ▶ During the execution the minimum execution duration is checked. If the value is not 0 an event is triggered after it expired independent of whether the phase still checks its phase done condition or only waits for the minimal execution duration.

Read more about execution duration in chapter Execution duration (auf Seite 35).

## 7.3 Status Line

The status line is automatically adapted to each arriving recipes and initialized with its data. The the status line the following is displayed:

- ▶ Error status of the recipe
  - Error: Currently there is an error in the recipe. The number of errors is also displayed.
  - Historical error: During the execution at least one error occurred. Currently not error exists.
  - No error: Until now the execution runs faultless.
- ▶ the current status of the REE (auf Seite 125) during the execution
- ▶ Mouse cursor position and changes such as moving the connection line
- ▶ the deleting, adding or selecting of elements

In the PFC editor the status line additionally offers the possibility in the right-hand area to display and control the zoom steps of the current view.

## 7.4 Recipe types and recipe states

Basically we distinguish between the following recipe types:

- ▶ PFC recipe for free, complex processes. They are created with the PFC editor (auf Seite 64).
- ▶ Matrix recipe for simple sequential/parallel processes. They are created with the matrix editor (auf Seite 101).

For both recipe types its true that they are available in the following characteristics:

- ▶ Master recipes:  
Form the basis for the control recipes. The created in status `Editable`. If they are in status `Released`, they can no longer be modified.
- ▶ Control recipes:  
Are recipes which can be executed once. Each control recipe is based on a master recipe. Control recipes can be modified concerning the process. The process is defined by the master recipe.

Changeable are only command tags for which option `Changeable in control recipe` was activated.

## STATUS FOR MASTER RECIPES

Master recipes can have the following states:

- ▶ **Editable:** In this state everything can be changed. Each new master recipe is created in this status.
- ▶ **Test mode (auf Seite 106):** In this status the recipe behaves similar as a control recipe. As them the process cannot be modified. It can be executed and all commands, actions, modes, etc. are available. You can also change all command tags for which option `Changeable in recipe` was activated.

Exception: If a phase is active at the moment, changes are not possible. Decisive for this is the status at clicking button OK. If the phase is active, the value changes are not taken over and an error message is displayed.

In test mode recipes can be executed consecutively several times. This is not the case for control recipes. Changes from the Editor are taken over after a reload or after a Runtime restart.

- ▶ **Released (auf Seite 108):** In this status no changes to the recipe are possible

**Note:** Also changes to the phases or to the reactions in the Editor are not transferred to a master recipe. The status at the release are frozen.

Exception: Changes to variable and function are not considered by these protection mechanisms. This can lead to a master recipe and all control recipes based on it becoming invalid. If e.g. the data type of a variable is changed from Bool to String, the validation function is no longer run through and the recipe can no longer be used as long as the error is not fixed in the Editor.

Likewise the units themselves are not protected as they are used recipe-spanning. If a unit name is changed in the Editor, it immediately takes effect after reloading or a Runtime restart on all phases which are based on it.

To be able to release a master recipe, the recipe must be validated without errors. The validation (auf Seite 123) is done automatically during the release and cannot be avoided. Control recipes can only be created from released recipes.

If a master recipe with status `Release` should be modified, you must create a copy of the master recipe via command `duplicate`. The copy gets status `Editable`.

## STATUS FOR CONTROL RECIPES

Control recipes can have the following states:

- ▶ **Prepared:** A newly created control recipe has this status. In this status it can be started and command tags, for which option `Changeable in control recipe` was activated, can be changed.
- ▶ **In execution:** The control recipe was started and is processed. It remains in this status until it is `Completed`, `Stopped` or `Canceled`. Via the **REE Status** (auf Seite 125) you can learn the exact state of the recipe.  
In this status command tags, for which option `Changeable in control recipe` was activated, can be changed.  
Exception: If a phase is active at the moment, changes are not possible. Decisive for this is the status at clicking button OK. If the phase is active, the value changes are not taken over and an error message is displayed.
- ▶ **Finished:** The recipe reached its final state. It can no longer be restarted and it also cannot be changed. Changes to command tags are not possible. Finished control recipes can be duplicated and deleted.

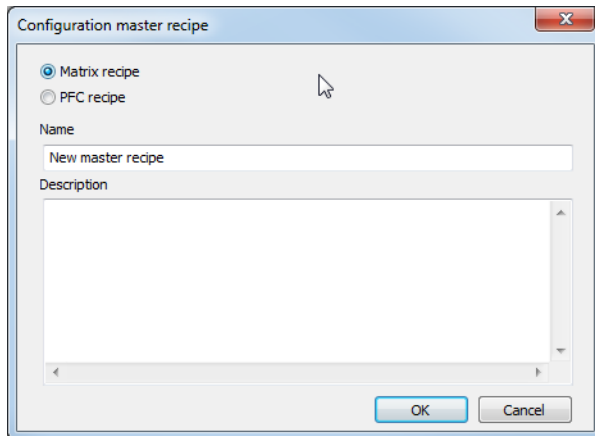
## 7.5 Master recipes

Master recipe are the basis of control recipes. With the help of master recipes you define and test the recipe process. After a master recipe is released, its content and structure can no longer be changed.

For the creation of master recipes two editors are available. Depending on the license only one of them or both can be used.

### 7.5.1 Create master recipe

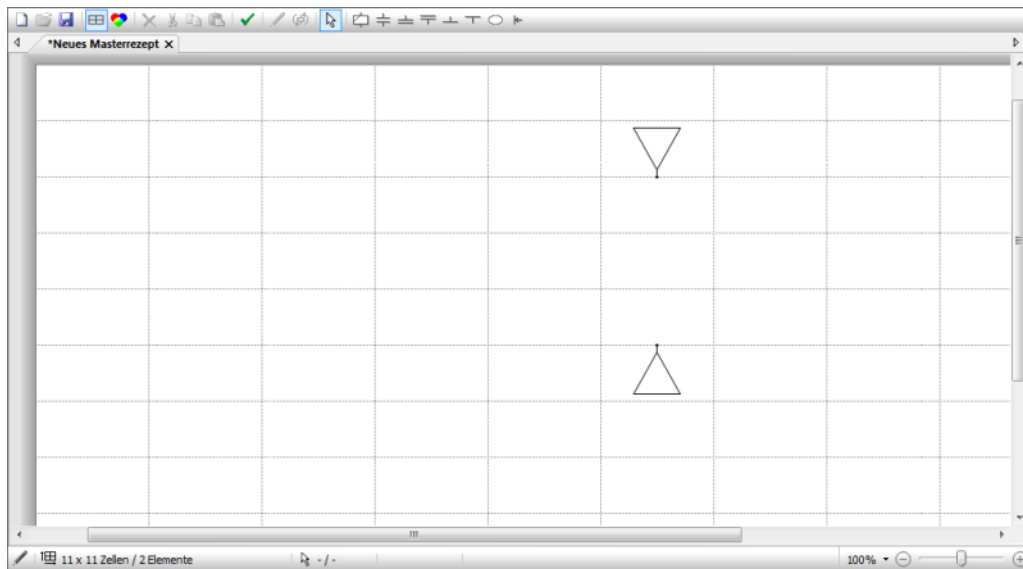
A click on button **Create master recipe** opens the dialog **Configuration master recipe**.



Tag	Description
Matrix recipe (auf Seite 100)	<p>Activate this radio button if you want to create a matrix recipe (auf Seite 100).</p> <p><b>Note:</b> Only possible if the corresponding license is available.</p>
PFC recipe (auf Seite 63)	<p>Activate this radio button if you want to create a PFC recipe (auf Seite 63).</p> <p><b>Note:</b> Only possible if the corresponding license is available.</p>
Name	<p>Unique name for the recipe. The name must not contain a dot (.), a @ or an asterisk (*).</p> <p>Maximum length: 256 characters.</p> <p>Note: When you copy a recipe the existing name is complemented 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.</p> <p>The uniqueness is checked in the entire network. Therefore it can happen that you cannot take over the name as another user on another computer in the zenon network already has used the same name and you do not see the recipe in the list of the master recipes yet.</p> <p>The recipe names can be changed later but only as long as the recipe is in status Editable.</p>
Description	<p>Optional description for the recipe which should be created.</p> <p>The description can be changed later but only as long as the recipe is in status Editable. To change the description select button <b>Rename master recipe</b>.</p>
OK	Applies all settings and created a new recipe.
Cancel	Closes the dialog without creating a recipe.

## 7.5.2 PFC recipe

If you selected PFC recipe in dialog **Configuration master recipe** and exited the dialog with OK, the newly created recipe opens on a new tab in the PFC editor (auf Seite 64).

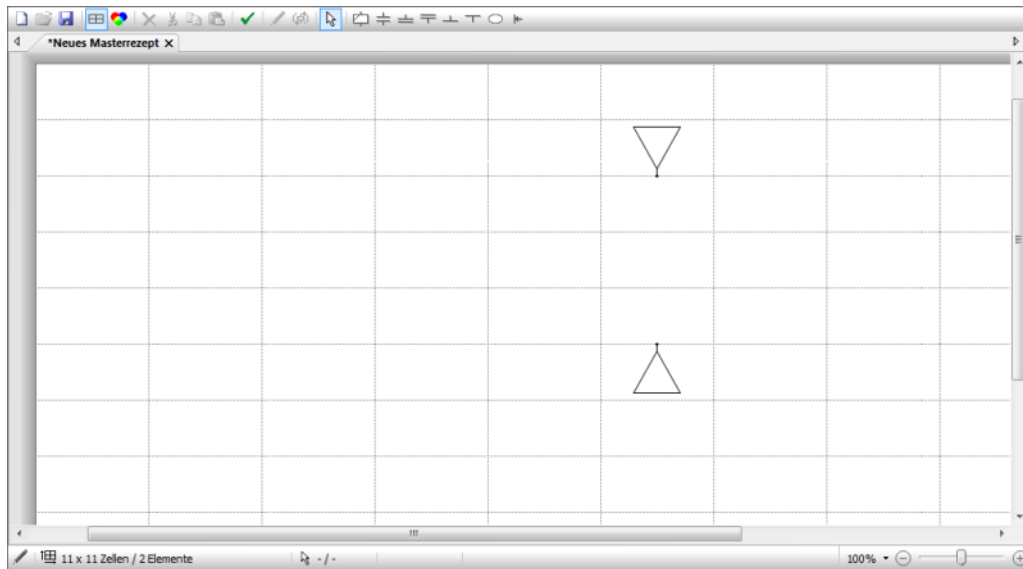


As each recipe needs a begin and an end element, these two elements already exist in the recipe and cannot be deleted from it.

Add the desired phases, transitions, branches, parallel branches and unit allocations to your recipe. Fields which are unsuitable for adding an element turn red when you move the element above it.

## PFC editor

In the PFC editor you can create your recipes graphically.



## TECHNICAL DETAILS

- ▶ Sheet size:
  - Default: 10 x 10 cells
  - Minimum: 5 x 5 cells
  - Maximum: 500 x 1000 cells
- ▶ Cell size
  - Default: 155 x 111 pixels
- ▶ Outside edge: 100 pixel
- ▶ Grid: is displayed by default; can also be hidden
- ▶ Scroll bar: Is displayed if the document is larger than the frame.
- ▶ Scrolling via mouse wheel: up and down or if you press and hold `Shift` left and right.
- ▶ Zooming: `Ctrl` + mouse wheel
- ▶ Selecting elements: left mouse click
- ▶ Multiple selection: `Ctrl` + mouse click



- **Move element:** 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.

## Tool bar and context menu PFC recipe

### TOOL BAR PFC EDITOR: EDIT MODE



Tag	Description
New master recipe...	Opens the dialog for creating a new master recipe.
Save master recipe	Saves the master recipe which is open for editing.  If another recipe is opened, the current recipe is saved automatically.
Show/hide grid	Shows/hides the grid.
Change background color and grid color.	Opens the dialog (auf Seite 69) for changing the color of the background and the grid.
Delete	Deletes the selected elements from the diagram.  Multiple selection via <code>Ctrl+mouse click</code> .
Check recipe for errors	Checks recipe for errors and displays found errors in an information window. For several errors the first errors are displayed.  The error message contains the error number, the ID of the element, its location and a message in plain text.
Edit element	Opens the corresponding dialog for editing the selected element.
Exchange phase	Opens dialog for selecting a phase and replaces the existing phase by the newly selected phase. You can select several phases and exchange them with a single phase.  Shortcut: <code>Shift+double click</code>
Edit mode	Toggles between insert mode and edit mode.
Add phase	Adds a phase.  Remains active as long as you change to the edit mode via symbol <b>Edit mode</b> or key <code>Esc</code> or you change to another insert option via a symbol for adding a new element.
Add transition	Adds a transition.  Remains active as long as you change to the edit mode via symbol <b>Edit mode</b> or key <code>Esc</code> or you change to another insert option via a symbol for adding a new element.
Add begin parallel branch	Adds a begin parallel branch.  Remains active as long as you change to the edit mode via symbol <b>Edit mode</b> or key <code>Esc</code> or you change to another

	insert option via a symbol for adding a new element.
<b>Add end parallel branch</b>	<p>Adds an end parallel branch.</p> <p>Remains active as long as you change to the edit mode via symbol <b>Edit mode</b> or key <b>ESC</b> or you change to another insert option via a symbol for adding a new element.</p>
<b>Add begin branch</b>	<p>Adds a begin branch.</p> <p>Remains active as long as you change to the edit mode via symbol <b>Edit mode</b> or key <b>ESC</b> or you change to another insert option via a symbol for adding a new element.</p>
<b>Add end branch</b>	<p>Adds an end branch.</p> <p>Remains active as long as you change to the edit mode via symbol <b>Edit mode</b> or key <b>ESC</b> or you change to another insert option via a symbol for adding a new element.</p>
<b>Add unit allocation</b>	<p>Adds a unit allocation.</p> <p>Remains active as long as you change to the edit mode via symbol <b>Edit mode</b> or key <b>ESC</b> or you change to another insert option via a symbol for adding a new element.</p>
<b>Add jump target</b>	<p>Adds a jump target.</p> <p>Remains active as long as you change to the edit mode via symbol <b>Edit mode</b> or key <b>ESC</b> or you change to another insert option via a symbol for adding a new element.</p>
<b>Switch recipe to test mode</b>	Switches recipe to the test mode. For this the recipe must be without errors.
<b>Release recipe</b>	Releases the recipe. With this a control recipe can be created.

## TOOL BAR PFC RECIPE: RELEASE



Tag	Description
<b>Check recipe for errors</b>	Checks recipe for errors and displays found errors in an information window. For several errors the first errors are displayed.  The error message contains the error number, the ID of the element, its location and a message in plain text.
<b>Edit element</b>	Opens the corresponding dialog for editing the selected element.
<b>Show/hide grid</b>	Shows/hides the grid.
<b>Change background color and grid color</b>	Opens the dialog (auf Seite 69) for changing the color of the background and the grid.
<b>Duplicate Recipe</b>	Only active if exactly one recipe is selected. Created a copy of the selected recipe. At the creation of the copy, the version of the recipe saved on the hard disk is used. If the recipe is just edited in another computer and the changes have not yet been saved, the changes are not applied. The dialog for the input of a unique name and the description is opened.
<b>Create control recipe</b>	Creates a control recipe.

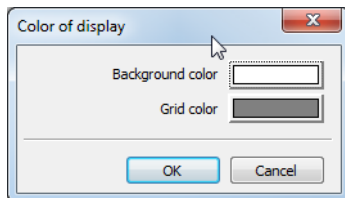
## CONTEXT MENU TABS

Each diagram is highlighted in the editor with a tab Via the context menu the following commands are available for all tabs:

Tag	Description
<b>Save</b>	Saves all changes in the diagram.
<b>Close</b>	Closes the diagram.
<b>Close all others</b>	Closes all diagrams except the diagram whose context menu is active.

## Settings diagram

A click on the symbol for the color configuration in the tool bar opens the dialog for configuring the color for the background and grid.



Tag	Description
Background color	Defines the background color of the diagram. Click on the color in order to open the palette for selecting a color.
Grid color	Defines the line color of the grid. Click on the color in order to open the palette for selecting a color.
OK	Applies all settings and closes the dialog.
Cancel	Discards all changes and closes the dialog.

## Begin element

First element in the process. It is automatically created in the editor when a recipe is created and it cannot be deleted.

## Allocate and configure unit

To allocate a unit:

1. in the tool bar select the symbol for **Add unit allocation**
2. move the mouse to the desired position
3. place the unit
4. the unit is added

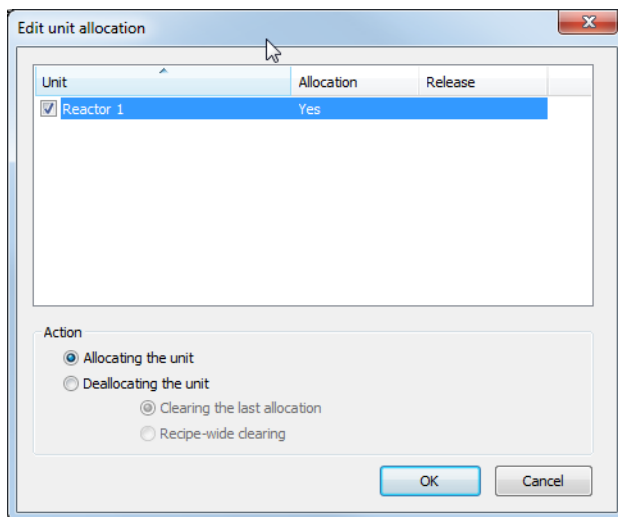
To configure the unit allocation:

1. double click the unit

- the dialog for the configuration is opened

## CONFIGURATION

In a configuration dialog you can select the units which exist in the batch recipe for the Runtime. It is distinguished between allocation and release. Releases can be set selectively and globally. Units which were selected before and no longer exist in the recipe, are displayed with an appropriate note.



Tag	Description
<b>List units</b>	Displays existing units and their assigned actions.
<b>Action</b>	Assigns an action to the unit selected in the list.
Allocation of the unit.	Allocates the unit.
Release of the unit.	Releases the unit: <ul style="list-style-type: none"> <li>▶ Release of the last allocation: The last allocation is released.</li> <li>▶ Recipe-spanning release: All allocations in the recipe are released.</li> </ul>
<b>OK</b>	Applies settings and closes the dialog.
<b>Cancel</b>	Discards changes and closes the dialog.

## ALLOCATE AND RELEASE UNITS

The allocation of a unit by element unit allocation or by a phase is only possible if the unit is not yet allocated or only in the same recipe. Allocation and release of units is always done in a cycle. It is always waited until all units which should be allocated are released. Then all allocations and releases are down

at the same time. Phases which are located in front of an element unit allocation remain active until the allocation was successful. When the recipe is finished and there are still allocations of elements `unit allocation` active, they are released implicitly.

**Note:** In states `Paused` or `Held`, units can also not be allocated by other recipes. But: In manual mode you can force the allocation of a unit by another recipe. The recipe with the first allocation keeps the control and takes priority at the execution. If this recipe withdraws its main allocation, the recipe with the longest active forced allocation takes over the main allocation.

## ACTIONS

For each unit which is used in the recipe, you can define an action:

- ▶ no action
- ▶ allocate
- ▶ release

## TOOLTIP

In the tooltip of element `unit allocation` all units are displayed which are marked for allocation or release. During the execution the Execution duration (auf Seite 35) is stated and all units for which you must wait are color-coded. You must wait for units if they are allocated in another recipe.

## Add and configure phase

To add a phase:

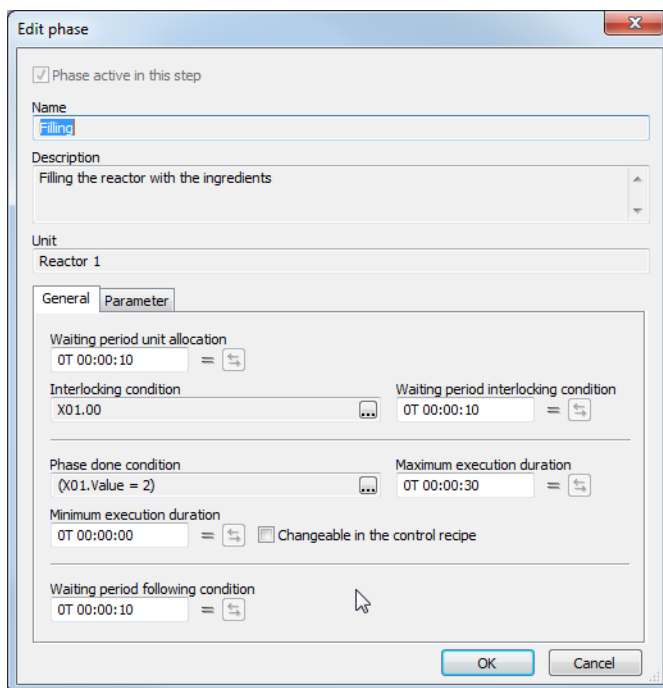
1. in the tool bar select the symbol for **Add phase**
2. move the mouse to the desired position
3. locate the phase
4. the dialog for selecting a phase (auf Seite 73) is opened
5. select the desired phase

To configure a phase:

1. double click the phase

- the dialog for the configuration is opened

## CONFIGURATION



Tag	Description
<b>Phase active in this step</b>	Active: Phase is active in this step
<b>Name</b>	Name of the phase. Only display.
<b>Description</b>	Comment about the phase. Only display.
<b>Unit</b>	Unit on which the phase is carried out. Only display.
<b>OK</b>	Applies all changes on all tabs and closes the dialog.
<b>Cancel</b>	Discards all changes on all tabs and closes the dialog.

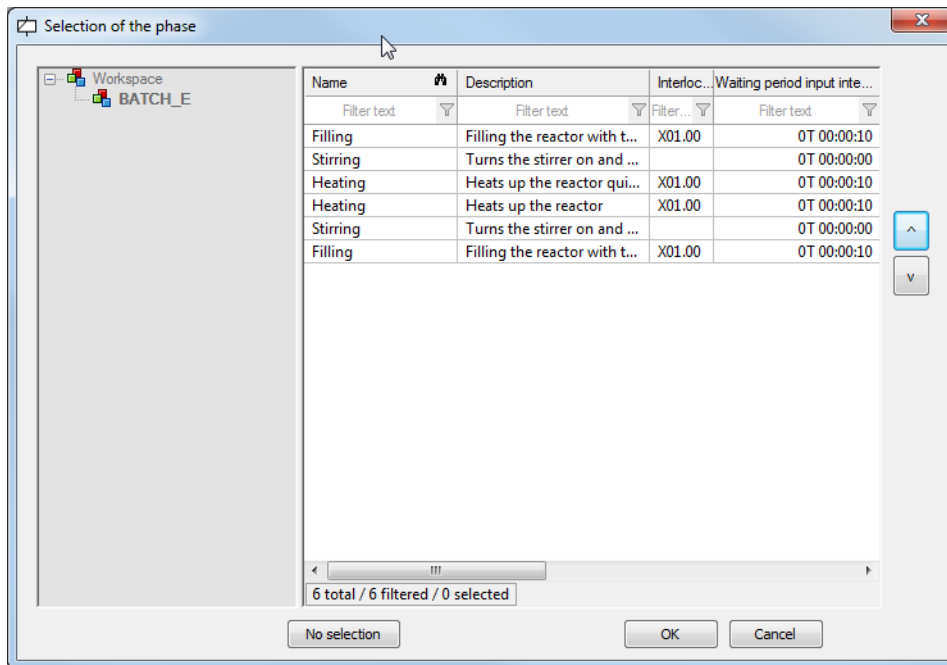
Configuration of the tabs see chapter:

- General (auf Seite 74): Display and configuration of the settings for the phase
- Tag (auf Seite 78): Configuration of the tags



## Selection phase

If a phase is added, the dialog for selecting a phase is opened.






Tag	Description
<b>Project tree</b>	Displays the current project from which the phases can be selected.
<b>List field phases</b>	<p>In the list all phases engineered in the Editor are displayed.</p> <p>This list can be filtered. The filtering is case-sensitive. Placeholders * and ? can be used.</p>
<b>Arrow keys</b>	Move selected phases up or down.
<b>No selection</b>	Deletes already selected phases from the element.
<b>OK</b>	Applies settings and closes the dialog.
<b>Cancel</b>	Discards all changes and closes the dialog.

All settings of the dialog are saved user-specificly on closing the dialog.

## General

The properties of tab **General** are set in the Editor with the exception of **Changeable** in the control recipe. The values can be changed in the master recipe. A symbol indicates whether the value in the dialog matches the value in the Editor. For different values you can again apply the value defined in the Editor.

Meaning of the symbols next to the values:

Tag	Description
 :	Value in the dialog and the value in the Editor match.
 :	Value in the dialog and the value in the Editor do not match.
 :	Only active if the values in the recipe and Editor do not match. Click on button to apply the value from the Editor. It overrides the value in the master recipe.

) are shown next to most values. The 'Changeable in the control recipe' checkbox is checked. OK and Cancel buttons are at the bottom." data-bbox="116 426 533 731"/>

For information about the base data on these tabs see chapter Add and configure phase (auf Seite 71).

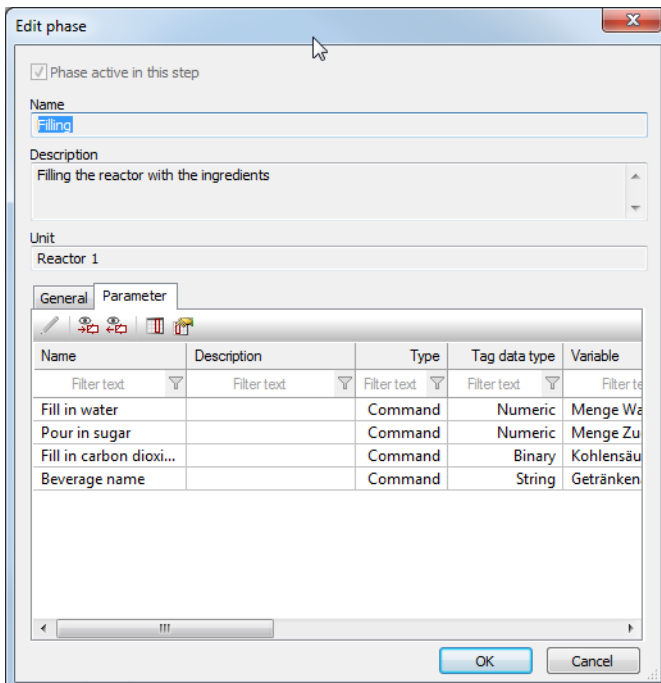
Tag	Description
Waiting period unit allocation	<p>Time in days, hours, minutes and seconds which is waited for the allocation of the unit. The waiting period can be changed in the Runtime as long as the recipe has status <code>Editable</code>.</p> <p>After the defined period has exceeded, the event <code>Waiting period unit allocation exceeded</code> is triggered and the element is highlighted. Additional actions must be defined by the engineer. If no further actions take place, the waiting is continued.</p> <p>Maximum: 9999d 23:59:59 Default: 0d 00:00:00</p> <p><b>Note:</b> A unit can only be allocated by a single recipe with status <code>In execution</code> at a time.</p> <p>Value is predefined in the Editor and can be changed here as long as the recipe is not released.</p>
Input interlocking	<p>Defines conditions for input interlocking. Click on button <code>...</code> or an entry to open the formula editor (auf Seite 139) for defining the condition.</p> <p>If an input interlocking is configured, the phase is only executed in the Runtime when the condition for the input interlocking is fulfilled. The formula can consist of one or more command tags and return tags of the phase. Value and status of the variables can be used. The formula returns <code>TRUE</code> or <code>FALSE</code> as result. The condition can be displayed in the Runtime but cannot be changed there.</p> <p>The waiting period for the input interlocking is configured with the help of property <code>Tempo di attesa per interblocco di ingresso</code>.</p>
Waiting period input interlocking	<p>Period in days, hours, minutes and seconds in which the condition defined in property <code>Interblocco di ingresso</code> must return value <code>TRUE</code>. The waiting period begins with the beginning of the check of the input interlocking. If the condition is not fulfilled within the waiting period, the event <code>Waiting period input interlocking exceeded</code> is triggered and the waiting is continued. If no reaction was defined for the event which forces another behavior, it is waited</p>

	<p>until the condition is fulfilled.</p> <p>If <code>0d 00:00:00</code> is defined as waiting period, the event is not triggered.</p> <ul style="list-style-type: none"> <li>▶ Minimum value: <code>0d 00:00:00</code></li> <li>▶ Maximum value: <code>9999d 23:59:59</code></li> <li>▶ Default: <code>0d 00:00:00</code></li> </ul> <p>Value is predefined in the Editor and can be changed here as long as the recipe is not released.</p>
Phase done condition	<p>Defines the condition for phase done. Click on button ... or an entry to open the formula editor (auf Seite 139) for defining the condition.</p> <p>The condition can be displayed in the Runtime but cannot be changed there. The period which is waited for the fulfillment of the condition is defined via property <code>Massima durata di esecuzione</code>.</p>
Maximum execution duration	<p>Period in days, hours, minutes and seconds in which the condition defined in property <code>Condizione di termine fase</code> must return value <code>TRUE</code>. The waiting period begins when writing of the command parameter. If the condition is not fulfilled within the waiting period, the event <code>Waiting period phase done condition exceeded</code> is triggered and the waiting is continued. If no reaction was defined for the event which forces another behavior, it is waited until the condition is fulfilled.</p> <p><b>Note:</b> Time keeps running when the recipe/the phase is paused.</p> <p>If <code>0d 00:00:00</code> is defined as waiting period, the event is not triggered.</p> <ul style="list-style-type: none"> <li>▶ Minimum value: <code>0d 00:00:00</code></li> <li>▶ Maximum value: <code>9999d 23:59:59</code></li> <li>▶ Default: <code>0d 00:00:00</code></li> </ul> <p>Value is predefined in the Editor and can be changed here</p>

	<p>as long as the recipe is not released.</p> <p>Makes only sense if property <code>Condizione di termine fase</code> was configured.</p> <p>Read more about execution duration in chapter Execution duration (auf Seite 35).</p>
Minimum execution duration	<p>Minimum execution duration of the phase.</p> <ul style="list-style-type: none"> <li>▶ <code>&gt;0</code>: Period which is at least waited after the writing of the command tag regardless of whether the phase done condition is fulfilled.</li> <li>▶ <code>0</code>: Execution duration is not checked</li> </ul> <p>Minimum execution duration can exceed <code>Massima durata di esecuzione</code>. Value is predefined in the Editor and can be changed here as long as the recipe is not released.</p> <p>Changeable in the control recipe if option <code>Changeable in the control recipe</code> was activated in the master recipe.</p> <p>Read more about execution duration in chapter Execution duration (auf Seite 35).</p>
Changeable in the control recipe	<p><code>Active</code>: Property can be changed in the control recipe.</p>
Waiting period following conditions	<p>Period in days, hours, minutes and seconds in which the phase must be completed. The waiting period begins when the phase done condition is reached. If the condition is not fulfilled within the waiting period, the event <code>Waiting period following condition exceeded</code> is triggered and the waiting is continued. If no reaction was defined for the event which forces another behavior, it is waited until the condition is fulfilled.</p> <p>If <code>0d 00:00:00</code> is defined as waiting period, the event is not triggered.</p> <ul style="list-style-type: none"> <li>▶ Minimum value: <code>0d 00:00:00</code></li> </ul>

	<p>► Maximum value: 9999d 23:59:59</p> <p>► Default: 0d 00:00:00</p> <p><b>Note:</b> If the following condition exists depends on the recipe structure. Therefore the configuration of a waiting period should not be done in the Editor but in the Runtime. Value is predefined in the Editor and can be changed here as long as the recipe is not released.</p> <p>For more information see chapters <b>Times</b> (auf Seite 22) and <b>Following conditions</b> (auf Seite 134).</p>
<b>OK</b>	Applies all changes on all tabs and closes the dialog.
<b>Cancel</b>	Discards all changes on all tabs and closes the dialog.

## Tag



**Edit phase**

☒ Phase active in this step

Name  
Filling

Description  
Filling the reactor with the ingredients

Unit  
Reactor 1

General Parameter

Name	Description	Type	Tag data type	Variable
Fill in water		Command	Numeric	Menge Wa
Pour in sugar		Command	Numeric	Menge Zu
Fill in carbon diox...		Command	Binary	Kohlensäu
Beverage name		Command	String	Getränken

OK Cancel

For information about the base data on these tabs see chapter Add and configure phase (auf Seite 71).

Tag	Description
List tag	Displays the tag configured in the Editor. Tags can be filtered and sorted according to columns.  Click on symbol <b>Edit tag</b> , double click the tag, menu item in the context menu or press <b>Return</b> to open the dialog (auf Seite 79) for editing a tag.
OK	Applies all changes on all tabs and closes the dialog.
Cancel	Discards all changes on all tabs and closes the dialog.

## TOOL BAR



Symbol	Meaning
<b>Edit tag</b>	Opens the dialog (auf Seite 79) for editing the tag.
<b>Display all command tags</b>	Toggles between the display of the changeable tags and all tags.
<b>Display return tag</b>	In addition to the command tags also displays the return tags or hides them.
<b>Column selection</b>	Opens the dialog (auf Seite 90) for selecting the columns which should be displayed.
<b>Column format</b>	Opens the dialog (auf Seite 91) for formatting the columns.

## Edit tag

To edit a tag in the Runtime:

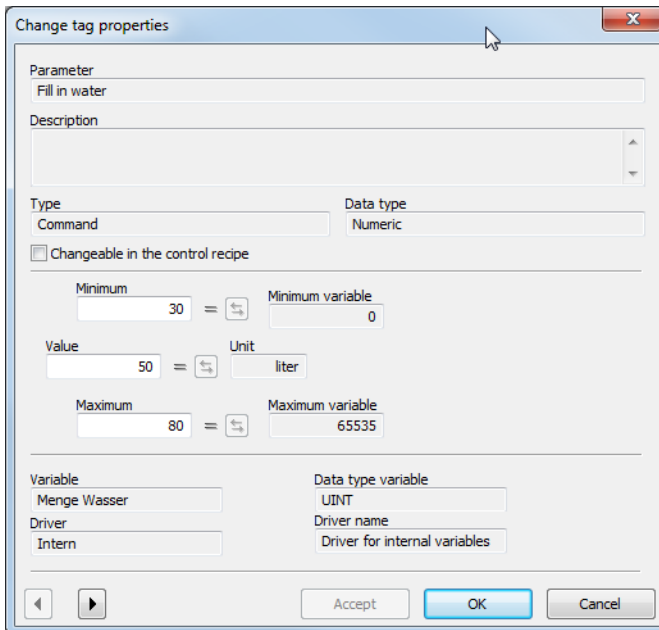
1. in dialog Edit phase select tab **Tag**
2. highlight the desired tag
3. open the dialog for editing the tags via a click on symbol **Edit tag**, the context menu, a double click on the tag or press **Return**
4. the dialog for editing is opened

For each data type an own dialog is opened:

- ▶ Numeric
- ▶ Binary
- ▶ String
- ▶ Duration

The properties are normally configured in the Editor and only displayed in the Runtime. Exceptions are values of the data type. They can be adapted if property `Modificabile` in `ricetta` was activated in the Editor.

## NUMERIC TAG



The screenshot shows the 'Change tag properties' dialog box for a numeric tag. The dialog has a title bar with a close button. The main content area is divided into several sections:

- Parameter:** A text field containing 'Fill in water'.
- Description:** A large text area with a vertical scrollbar.
- Type:** A dropdown menu showing 'Command'.
- Data type:** A dropdown menu showing 'Numeric'.
- Changeable in the control recipe:** A checkbox that is currently unchecked.
- Range and Units:** A section with three rows of input fields and labels:
  - Minimum: 30, Minimum variable: 0
  - Value: 50, Unit: liter
  - Maximum: 80, Maximum variable: 65535
- Variable:** A text field containing 'Menge Wasser'.
- Data type variable:** A dropdown menu showing 'UINT'.
- Driver:** A text field containing 'Intern'.
- Driver name:** A text field containing 'Driver for internal variables'.

At the bottom of the dialog, there are three buttons: 'Accept', 'OK' (highlighted in blue), and 'Cancel'. There are also navigation arrows on the left side of the bottom bar.



Tag	Meaning
Tag	Name of the tag. Only display.
Description	Free description of the tag. Only display.
Type	Type of the tag: Command or Return. Only display.
Data type	Data type of the tag. Only display.
Changeable in the control recipe	<p>Active : Value of the property can be changed in the control recipe.</p> <p>Only available if:</p> <ul style="list-style-type: none"> <li>▶ the tag is of type command</li> <li>▶ it was configured in the Editor that the tag can be changed in the Batch recipe (property: <code>Modificabile in ricetta</code>).</li> </ul>
Minimum	<p>Minimum value for the tag.</p> <p>Default from the Editor can be changed when property <code>Modificabile in ricetta</code> was activated in the Editor.</p> <p>If the value is changed, it is marked with the symbol (auf Seite 74) on the right side of the value. Changed values can be overwritten with the default from the Editor with the help of the button right next to it.</p>
Minimum variable	Allowed minimum value of the variable.
Value	<p>Name of the tag.</p> <p>Default from the Editor can be changed when property <code>Modificabile in ricetta</code> was activated in the Editor.</p> <p>If the value is changed, it is marked with the symbol (auf Seite 74) on the right side of the value. Changed values can be overwritten with the default from the Editor with the help of the button right next to it.</p>
Unit	Unit of the value.
Maximum	<p>Maximum value for the tag. Only display.</p> <p>Default from the Editor can be changed when property</p>

	<p>Modificabile in ricetta was activated in the Editor.</p> <p>If the value is changed, it is marked with the symbol (auf Seite 74) on the right side of the value. Changed values can be overwritten with the default from the Editor with the help of the button right next to it.</p>
Maximum variable	Allowed maximum value of the variable. Only display.
Variable	Variable which is linked to the tag. Only display.
Data type variable	Data type of the variable. Only display.
Driver	Driver of the variable. Only display.
Driver description	Description of the driver of the variable. Only display.
Arrow keys	<p>Navigating through the tags.</p> <p>They are displayed in the order of the list. At this only tags are displayed which are visible with the current filter and grouping.</p> <p>If changes were done, there is a prompt before you can change to another tag whether the changes should be applied or discarded. If changes should be applied, the input is checked before advancing.</p>
<b>Apply</b>	Applies all changes if the check of the changes was successful. The dialog remains open for further editing.
<b>OK</b>	Applies all changes and closes the dialog if the check of the changes was successful.
<b>Cancel</b>	Discards all changes which have not been taken over yet and closes the dialog.

## BINARY TAG

Change tag properties


Parameter  
Fill in carbon dioxide

Description

Type  
Command

Data type  
Binary

☐ Changeable in the control recipe

☒ Off    =    

☐ On

☐ Toggle

Variable  
Kohlensäure zufüllen

Data type variable  
BOOL

Driver  
Intern

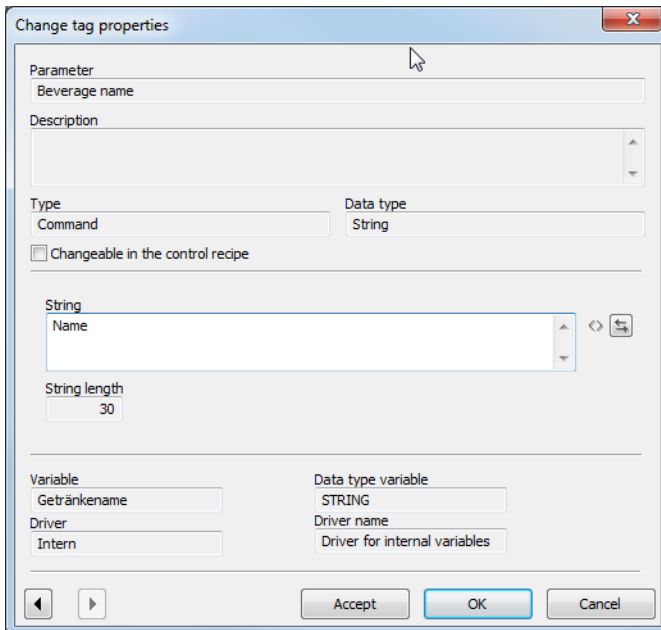
Driver name  
Driver for internal variables

Accept OK Cancel

Tag	Meaning
Tag	Name of the tag. Only display.
Description	Free description of the tag. Only display.
Type	Type of the tag: Command or Return. Only display.
Data type	Data type of the tag. Only display.
Changeable in the control recipe	<p>Active : Value of the property can be changed in the control recipe.</p> <p>Only available if:</p> <ul style="list-style-type: none"> <li>▶ the tag is of type command</li> <li>▶ it was configured in the Editor that the tag can be changed in the Batch recipe (property: <code>Modificabile in ricetta</code>).</li> </ul>
Off	<p>Status: Off.</p> <p>Default from the Editor can be changed when property <code>Modificabile in ricetta</code> was activated in the Editor.</p> <p>If the value is changed, it is marked with the symbol (auf Seite 74) on the right side of the value. Changed values can be overwritten with the default from the Editor with the help of the button right next to it.</p>
On	Status: On.
Toggle	Toggles between the states.
Variable	Variable which is linked to the tag. Only display.
Data type variable	Data type of the variable. Only display.
Driver	Driver of the variable. Only display.
Driver description	Description of the driver of the variable. Only display.
Arrow keys	<p>Navigating through the tags.</p> <p>They are displayed in the order of the list. At this only tags are displayed which are visible with the current filter and grouping.</p> <p>If changes were done, there is a prompt before you can change</p>

	to another tag whether the changes should be applied or discarded. If changes should be applied, the input is checked before advancing.
<b>Apply</b>	Applies all changes if the check of the changes was successful. The dialog remains open for further editing.
<b>OK</b>	Applies all changes and closes the dialog if the check of the changes was successful.
<b>Cancel</b>	Discards all changes which have not been taken over yet and closes the dialog.

## STRING TAG



The screenshot shows the 'Change tag properties' dialog box for a String tag. The dialog has a title bar with a close button. The main content area is divided into several sections:

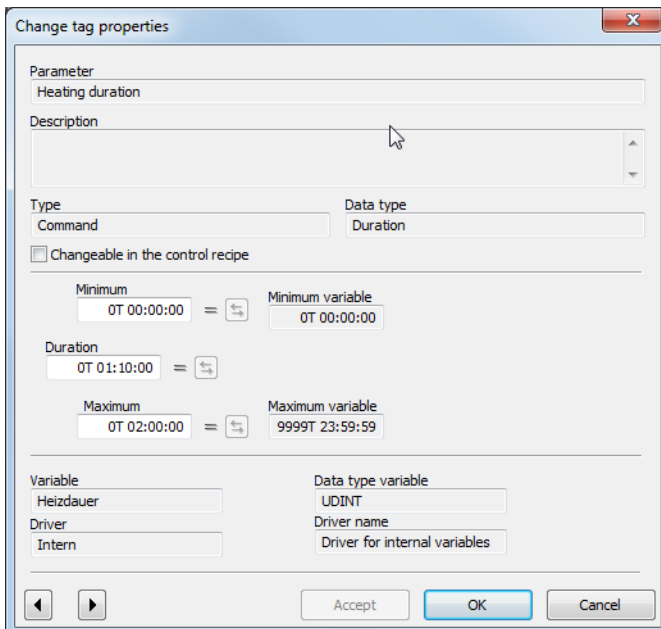
- Parameter:** A text field containing 'Beverage name'.
- Description:** A large text area with a vertical scrollbar.
- Type:** A dropdown menu showing 'Command'.
- Data type:** A dropdown menu showing 'String'.
- Changeable in the control recipe:** A checkbox that is currently unchecked.
- String:** A text field containing 'Name'.
- String length:** A text field containing '30'.
- Variable:** A text field containing 'Getränkename'.
- Data type variable:** A dropdown menu showing 'STRING'.
- Driver:** A text field containing 'Intern'.
- Driver name:** A text field containing 'Driver for internal variables'.

At the bottom of the dialog, there are three buttons: 'Accept', 'OK', and 'Cancel'. The 'OK' button is highlighted with a blue border.

Tag	Meaning
Tag	Name of the tag. Only display.
Description	Free description of the tag. Only display.
Type	Type of the tag. Command or Return. Only display.
Data type	Data type of the tag. Only display.
Changeable in the control recipe	<p>Active : Value of the property can be changed in the control recipe.</p> <p>Only available if:</p> <ul style="list-style-type: none"> <li>▶ the tag is of type command</li> <li>▶ it was configured in the Editor that the tag can be changed in the Batch recipe (property: <code>Modificabile in ricetta</code>).</li> </ul>
String	<p>Alphanumeric character string.</p> <p>Default from the Editor can be changed when property <code>Modificabile in ricetta</code> was activated in the Editor.</p> <p>If the value is changed, it is marked with the symbol (auf Seite 74) on the right side of the value. Changed values can be overwritten with the default from the Editor with the help of the button right next to it.</p> <p>Possible length is limited by the <code>Lunghezza stringa</code> engineered in the variable.</p>
String length	Defines possible length of the string. Only display.
Variable	Variable which is linked to the tag. Only display.
Data type variable	Data type of the variable. Only display.
Driver	Driver of the variable. Only display.
Driver description	Description of the driver of the variable. Only display.
Arrow keys	<p>Navigating through the tags.</p> <p>They are displayed in the order of the list. At this only tags are displayed which are visible with the current filter and grouping.</p>

	If changes were done, there is a prompt before you can change to another tag whether the changes should be applied or discarded. If changes should be applied, the input is checked before advancing.
<b>Accept</b>	Applies all changes if the check of the changes was successful. The dialog remains open for further editing.
<b>OK</b>	Applies all changes and closes the dialog if the check of the changes was successful.
<b>Cancel</b>	Discards all changes which have not been taken over yet and closes the dialog.

## DURATION



The screenshot shows the 'Change tag properties' dialog box for the 'Heating duration' parameter. The dialog is titled 'Change tag properties' and has a close button (X) in the top right corner. The 'Parameter' field contains 'Heating duration'. The 'Description' field is empty. The 'Type' is set to 'Command' and the 'Data type' is set to 'Duration'. There is a checkbox labeled 'Changeable in the control recipe' which is currently unchecked. Below this, there are three rows of time-related settings, each with a value and a variable assignment:

- Minimum: 0T 00:00:00 = Minimum variable: 0T 00:00:00
- Duration: 0T 01:10:00 =
- Maximum: 0T 02:00:00 = Maximum variable: 9999T 23:59:59

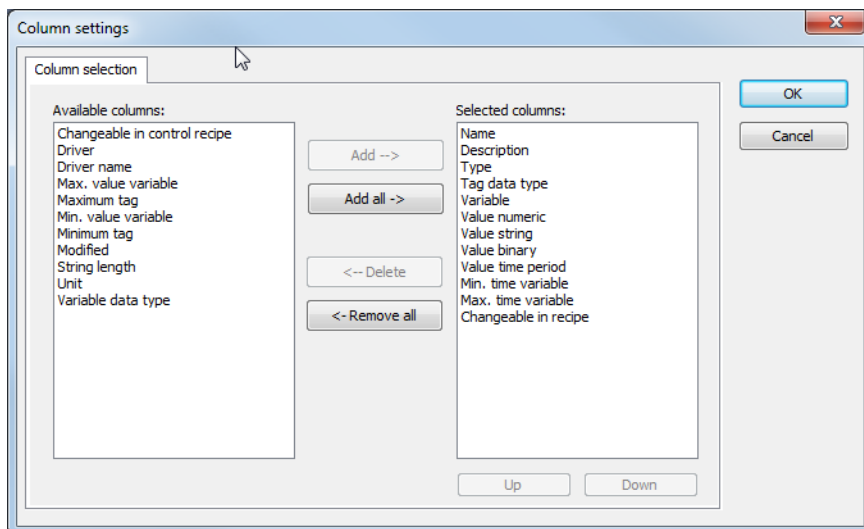
At the bottom, there are fields for 'Variable' (Heizdauer), 'Data type variable' (UDINT), 'Driver' (Intern), and 'Driver name' (Driver for internal variables). The 'Accept', 'OK', and 'Cancel' buttons are at the bottom right, with 'OK' highlighted in blue. There are also navigation arrows (back and forward) on the bottom left.

Tag	Meaning
Tag	Name of the tag. Only display.
Description	Free description of the tag. Only display.
Type	Type of the tag. Command or Return. Only display.
Data type	Data type of the tag. Only display.
Changeable in the control recipe	<p>Active : Value of the property can be changed in the control recipe.</p> <p>Only available if:</p> <ul style="list-style-type: none"> <li>▶ the tag is of type command</li> <li>▶ it was configured in the Editor that the tag can be changed in the Batch recipe (property: <code>Modificabile in ricetta</code>).</li> </ul>
Minimum	<p>Minimum value for the tag in format: <b>d hh:mm:ss</b></p> <p>Default from the Editor can be changed when property <code>Modificabile in ricetta</code> was activated in the Editor.</p> <p>If the value is changed, it is marked with the symbol (auf Seite 74) on the right side of the value. Changed values can be overwritten with the default from the Editor with the help of the button right next to it.</p>
Minimum variable	Allowed minimum value of the variable.
Duration	<p>Value of the tag in format: <b>d hh:mm:ss.</b></p> <p>Default from the Editor can be changed when property <code>Modificabile in ricetta</code> was activated in the Editor.</p> <p>If the value is changed, it is marked with the symbol (auf Seite 74) on the right side of the value. Changed values can be overwritten with the default from the Editor with the help of the button right next to it.</p>
Unit	Unit of the value.
Maximum	Maximum value for the tag in format:



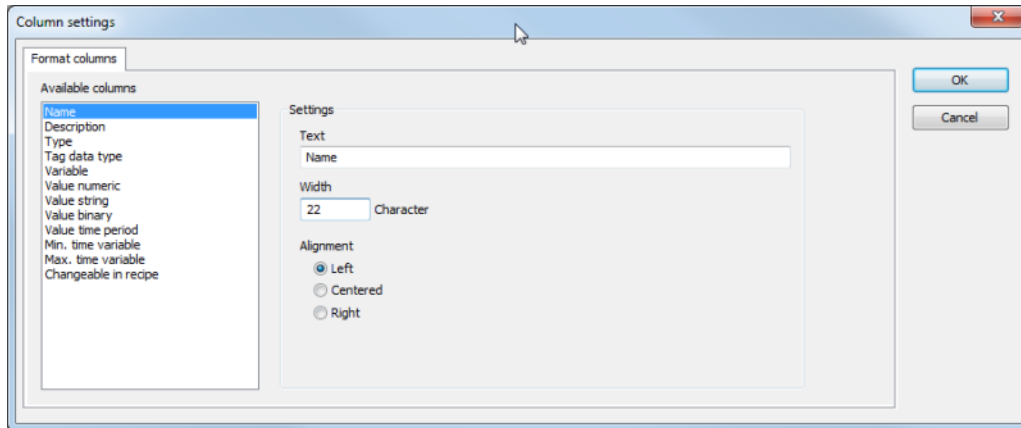
	<p><b>d hh:mm:ss.</b></p> <p>Default from the Editor can be changed when property <code>Modificabile in ricetta</code> was activated in the Editor.</p> <p>If the value is changed, it is marked with the symbol (auf Seite 74) on the right side of the value. Changed values can be overwritten with the default from the Editor with the help of the button right next to it.</p>
Maximum variable	Allowed minimum value of the variable. Only display.
Variable	Variable which is linked to the tag. Only display.
Data type variable	Data type of the variable. Only display.
Driver	Driver of the variable. Only display.
Driver description	Description of the driver of the variable. Only display.
Arrow keys	<p>Navigating through the tags.</p> <p>They are displayed in the order of the list. At this only tags are displayed which are visible with the current filter and grouping.</p> <p>If changes were done, there is a prompt before you can change to another tag whether the changes should be applied or discarded. If changes should be applied, the input is checked before advancing.</p>
<b>Apply</b>	Applies all changes if the check of the changes was successful. The dialog remains open for further editing.
<b>OK</b>	Applies all changes and closes the dialog if the check of the changes was successful.
<b>Cancel</b>	Discards all changes which have not been taken over yet and closes the dialog.

## Column selection



Pulsanti	Funzione
<b>Aggiungi</b>	Sposta le colonne selezionate da quelle disponibili a quelle selezionate. Dopo aver confermato il dialogo cliccando su OK, esse vengono visualizzate nella visualizzazione dettagliata.
<b>Aggiungi tutte</b>	Sposta tutte le colonne disponibili nelle colonne selezionate.
<b>Rimuovi</b>	Elimina le colonne evidenziate dalle selezionate e le visualizza nella lista di quelle disponibili. Dopo aver confermato il dialogo cliccando su OK, esse vengono eliminate dalla visualizzazione dettagliata.
<b>Cancella tutte</b>	Cancella tutti le colonne dalla lista delle colonne scelte.
<b>In sù</b>	Sposta l'inserimento selezionato verso l'alto. Questa funzione è sempre disponibile per inserimenti singoli, non è possibile però nel caso di una selezione multipla.
<b>In giù</b>	Sposta l'inserimento selezionato verso il basso. Questa funzione è sempre disponibile per inserimenti singoli, non è possibile però nel caso di una selezione multipla.
<b>OK</b>	Riprende le impostazioni e chiude il dialogo.
<b>Annulla</b>	Rifiuta le impostazioni e chiude il dialogo.
<b>Guida</b>	Apri la guida online

## Column format



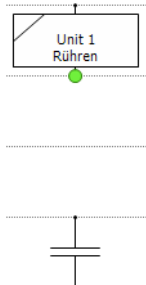
Tag	Description
<b>Available columns</b>	List of the available columns via <b>Column selection</b> . The column selected here is configured via settings of section <b>Property</b> .
<b>Tag</b>	Settings for selected column.
Labeling	Name for column title. The column title is online language switchable. For this you must enter the @ character in front of the name.
Width	Width of the column in characters. Calculation: Number times average character width of the selected font.
Alignment	Alignment.  Possible settings: <ul style="list-style-type: none"> <li>▶ Left-aligned: Text is aligned on the left border of the column.</li> <li>▶ Centered: Text is displayed centered in the column.</li> <li>▶ Right-aligned: Text is aligned on the right border of the column.</li> </ul>
<b>OK</b>	Applies settings and closes the dialog.
<b>Cancel</b>	Discards settings and closes the dialog.

## Lines

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

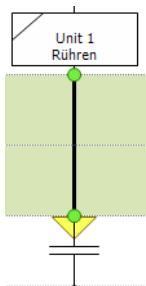
1. Activate a point with the help of the mouse.

The connection point turns green. Red means that the connection point is already taken.

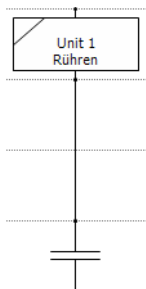


2. Drag a line to another connection point:

A yellow arrow shows the direction of the line. Green fields can be crossed. Red fields may not be crossed by the line.



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



## OPERATIONS WITH LINES

Lines:

- ▶ are dragged with the mouse
- ▶ can be moved (press and hold key `Ctrl`)

At this all existing connections are separated and it is tried to reconnect the line if there are

objects with connections points in the right direction at the target.

If several lines are highlighted, the line, in whose cell the mouse cursor is, is moved.

- ▶ can be deleted by highlighting them and pressing `Del`
- ▶ are deleted when re-dragging them from beginning 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

## Transition

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

For details about transitions see section Engineering in the Editor (auf Seite 12) in chapter Transitions (auf Seite 20).

## Branches

Recipes can branch<sup>1</sup> and run in parallel branches (auf Seite 96).

---

<sup>1</sup> A branch offers the possibility to execute one of several possible ways. For this it is necessary that the first element at the beginning of a branch is a transition. This means that Begin branch can only be followed by a transition. Procedure: The path is chosen for which the transition is TRUE first. 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 fo ...

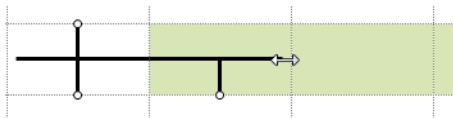
Branches and parallel branches consist of:

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

## ADD BRANCHES

To create a branch:

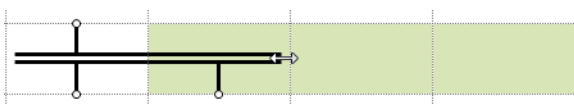
1. select the symbol **Add begin branch**
2. put the branch on the desired location
3. connect the input connection point with a output connection point of the preceding object
4. connect both output connection points with the desired following objects
5. close a branch with object **Add end branch**



## ADD PARALLEL BRANCH

To create a parallel branch:

1. select the symbol **Add 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 **Add end parallel branch**



## 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. keep the mouse button pressed
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 cursor over the object until it turns into a double arrow
2. press and hold the left mouse button and move the mouse in the desired direction:
  - away from the object to extend it
  - into the object to shorten it
3. at extending a new connection piece is added;
  - all fields which are concerned by the extension are marked green
  - to add several need connection pieces the process must be repeated
4. at shortening all corresponding connection pieces are deleted

## Branches

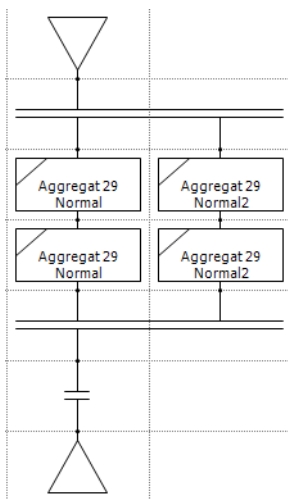
A branch offers the possibility to execute one of several possible ways. For this it is necessary that the first element at the beginning of a branch is a transition. This means that **Begin branch** can only be followed by a transitions (auf Seite 20).

Procedure:

- ▶ The path is chosen for which the transition is **TRUE** first.
- ▶ 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 phase in front of the element and a transition behind, the phase remains active until the transition was completed.

In a branch the objects are processed sequentially. Each branch processes its objects independent of other branches.



## Parallel branch

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.

In the process the respective intermediate area of the **end parallel branch** is also colored. The color matches the coloring (auf Seite 124) of the phase.

**Phase completed** is displayed as active as soon as the first previous element has been completed. This means that a phase is **Finished** or a transition is passed. Transitions are marked as completed as soon as they are passed. Phases wait at **Phase completed** until the **end parallel branch** is completed. Completed means that either the following phase is active or the following transition is inactive.

During the execution the status is color-coded.



## Split up and combine branches

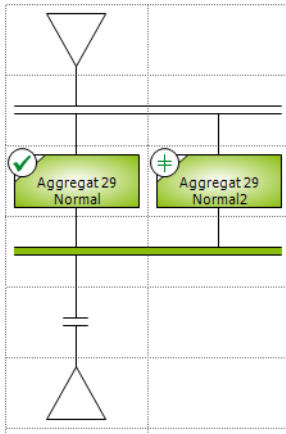
For parallel branches the branch splits up a **Begin parallel branch** and combines the single branches at **End parallel branch**. The paths of the parallel objects are independent branches. Only at **End parallel branch** all branches are synchronized.

The possible branches are defined by the engineering. If the separate branches are allocated or released, arises in the Runtime. A branch is active as long as an object on it is active.

The object types **Begin branch**, **End branch** and **Jump target** do not allocate and release branches as these objects are processed in the same branch. Combining branches is not allowed.

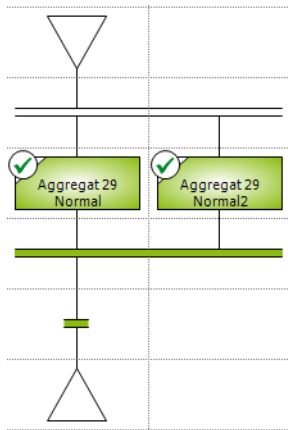
### END PARALLEL BRANCH

An **End parallel branch** combines the branches of the connected paths. The object after **End parallel branch** is activated when all paths reaches **End parallel branch** with their process.



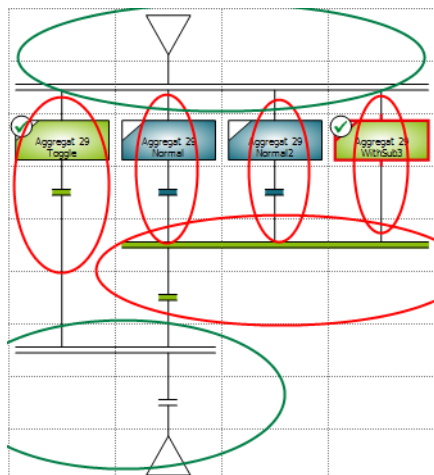
The left path is ready. Therefore **End parallel branch** is already active. The transition after **End parallel branch** is not yet active as the right branch is not yet completed.

When the right branch is also completed:



The transition after **End parallel branch** is activated. All objects which were active before are still active. Instead of the transition there could also be another **End parallel branch**.

### A LITTLE MORE COMPLEX ALTERNATIVE:



The areas highlighted in green are a branch.

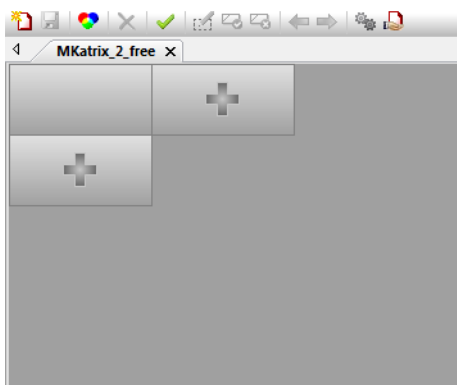


## End element

Last element in the process. It is automatically created in the editor when a recipe is created and it cannot be deleted.

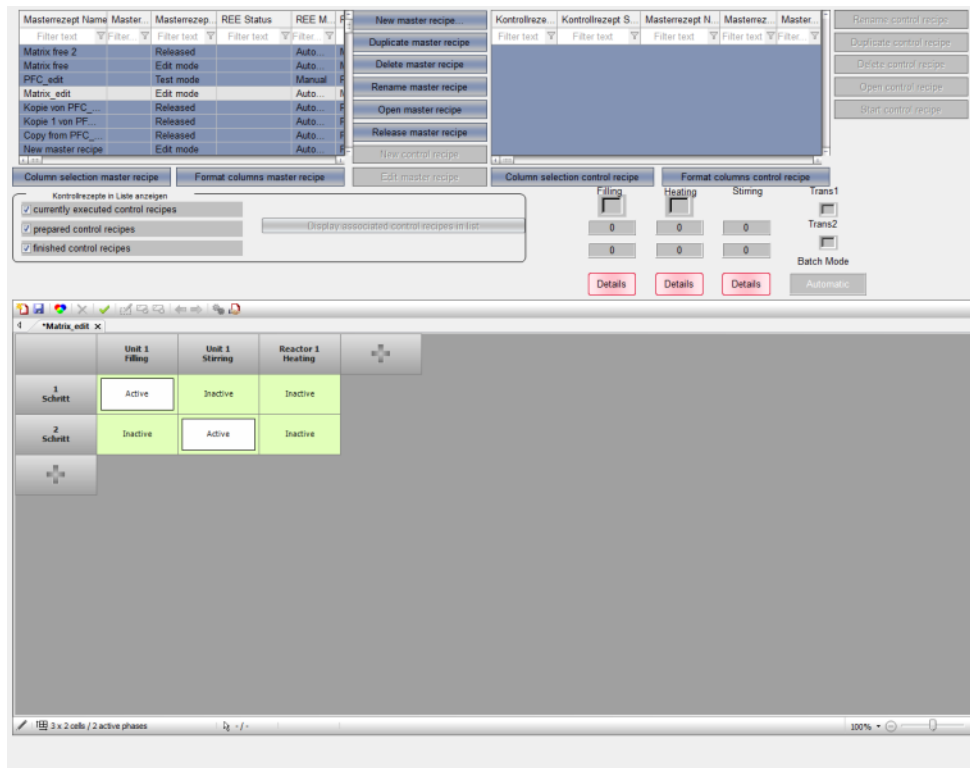
### 7.5.3 Matrix recipe

If you selected matrix recipe in dialog **Configuration master recipe** and exited the dialog with OK, the newly created recipe opens on a new tab in the matrix editor (auf Seite 101).



## Matrix editor

In the matrix editor you can create your recipes in a matrix.



## TECHNICAL DETAILS

- ▶ **Matrix:**
  - Columns contain phases.
  - Lines contain steps with active/inactive phases.
- ▶ **Add line/column:** Click on the plus sign to add a new line or column to the matrix. For new columns the dialog (auf Seite 104) for configuring new phases is opened automatically.
- ▶ **Delete line/column:** Press **Del** to delete the highlighted line or column.
- ▶ **Move lines/columns:** Lines and columns can be moved via drag & drop with the help of the mouse. Individual labels of steps remain; the step number is adapted automatically.
- ▶ **Border cells:**
  - Double click on phase: opens the dialog (auf Seite 104) for configuring the phase.

- Double click on step: opens the dialog for labeling (auf Seite 104) the step.
- ▶ Selection of lines/columns:
  - left mouse click in border cell: selects a line/column.
  - Ctrl+mouse click in border cell: selects additional lines/columns.
- ▶ Cells:
  - left mouse click: selects empty cell.
  - Shift + click: activates/deactivates phase.
  - Ctrl+click: selects several cells.
  - Double click on cell: opens the dialog for configuring (auf Seite 104) the phase.
- ▶ Scroll bar: Is displayed if the document is larger than the frame.
- ▶ Scrolling via mouse wheel: up and down or if you press and hold Shift left and right.

## Tool bars matrix recipe

### MATRIX RECIPE EDIT MODE



<b>New master recipe...</b>	Opens the dialog for creating a new master recipe.
<b>Save master recipe</b>	Saves the master recipe which is open for editing.  If another recipe is opened, the current recipe is saved automatically.
<b>Change background color and grid color</b>	Opens the dialog (auf Seite 69) for changing the color of the background and the grid.
<b>Delete</b>	Deletes the selected elements from the diagram.  Multiple selection via <code>Ctrl+mouse click</code> .
<b>Check recipe for errors</b>	Checks recipe for errors and displays found errors in an information window. For several errors the first errors are displayed.  The error message contains the error number, the ID of the element, its location and a message in plain text.
<b>Edit element</b>	Opens the corresponding dialog for editing the selected element.
<b>Activate selected phase</b>	Activates the selected phase.
<b>Deactivate selected phase</b>	Deactivates the selected phase.
<b>Move selected column to the left or selected step up</b>	Moves the selected column to the left by one position or moves selected step up by one position.
<b>Move selected column to the right or selected step down</b>	Moves the selected column to the right by one position or moves selected step down by one position.
<b>Switch recipe to test mode</b>	Switches recipe to the test mode. For this the recipe must be without errors.
<b>Release recipe</b>	Releases the recipe. With this a control recipe can be created.

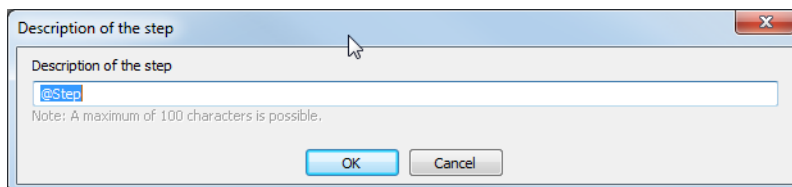
## RELEASE MATRIX RECIPE



Tag	Description
Check recipe for errors	Checks recipe for errors and displays found errors in an information window. For several errors the first errors are displayed.  The error message contains the error number, the ID of the element, its location and a message in plain text.
Edit element	Opens the corresponding dialog for editing the selected element.
Show/hide grid	Shows/hides the grid.
Change background color and grid color	Opens the dialog (auf Seite 69) for changing the color of the background and the grid.
Duplicate Recipe	Only active if exactly one recipe is selected. Created a copy of the selected recipe. At the creation of the copy, the version of the recipe saved on the hard disk is used. If the recipe is just edited in another computer and the changes have not yet been saved, the changes are not applied. The dialog for the input of a unique name and the description is opened.
Create control recipe	Creates a control recipe.

## Name steps

Steps in the matrix editor can be named individually. Double click on the cell to open the dialog for entering an individual name.



The name can have up to 100 characters and is language switchable is preceded by a @.

## Add and configure phase

To add a phase:

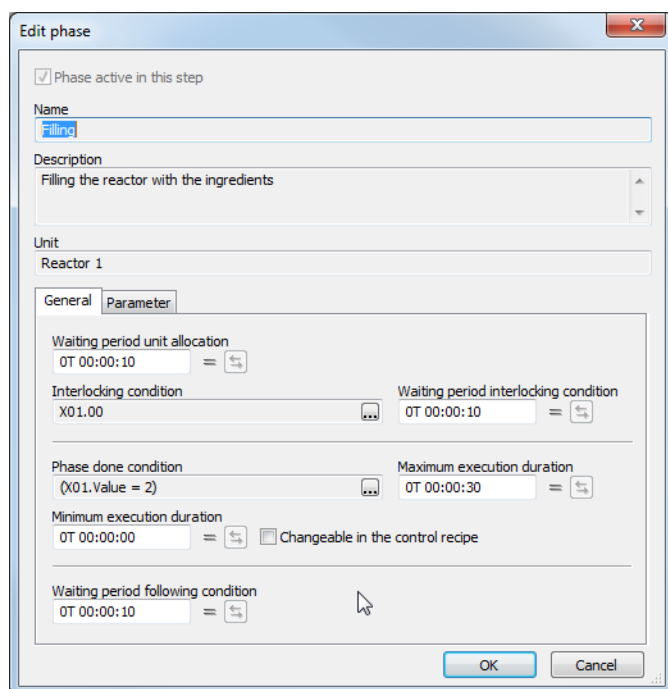


1. click on the plus sign in the last column of the matrix
2. the dialog for selecting a phase is opened
3. select the desired phase

To configure a phase:

1. double click the matrix field or the column with the desired phase
2. the dialog for the configuration is opened

## CONFIGURATION



The 'Edit phase' dialog box is shown with the following configuration:

- ☒ Phase active in this step
- Name:
- Description:
- Unit:
- General tab selected
- Waiting period unit allocation:  = [clock icon]
- Interlocking condition:  ...
- Waiting period interlocking condition:  = [clock icon]
- Phase done condition:  ...
- Maximum execution duration:  = [clock icon]
- Minimum execution duration:  = [clock icon] ☐ Changeable in the control recipe
- Waiting period following condition:  = [clock icon]
- Buttons: OK, Cancel

Tag	Description
Phase active in this step	Active: Phase is active in this step
Name	Name of the phase. Only display.
Description	Comment about the phase. Only display.
Unit	Unit on which the phase is carried out. Only display.
OK	Applies all changes on all tabs and closes the dialog.
Cancel	Discards all changes on all tabs and closes the dialog.

Configuration of the tabs see section PFC editor (auf Seite 64)/chapter:

- ▶ General (auf Seite 74): Display and configuration of the settings for the phase
- ▶ Tag (auf Seite 78): Configuration of the tags

#### 7.5.4 Master recipe - test mode

The test mode is used to test master recipes without releasing it and creating control recipes. In addition in the test mode changes in the Editor can be applied directly via reloading the Runtime.

**Exception:** During the execution of a recipe, the reloading of a recipe is delayed. Not until the recipe is finished, stopped or aborted, the reloading process is executed.

In test mode you cannot can the principle recipe process. You can only change values of the command tags. Changes are directly saved in the master recipe. It is not necessary to save explicitly.



Tag	Description
Start recipe	Starts the recipe process.
Pause recipe	Pauses the recipe.
Resume recipe	Resumes paused recipe.
Hold recipe	Holds recipe.
Restart recipe	Restarts held recipe.
Stop recipe	Stops the recipe.
Abort recipe	Aborts the recipe process.
Pause phase	Pauses the phase.
Resume phase	Resumes the process of a paused phase.
Hold phase	Holds phase.
Restart phase	Restarts held phase.
Check recipe for errors	<p>Checks recipe for errors and displays found errors in an information window. For several errors the first errors are displayed.</p> <p>The error message contains the error number, the ID of the element, its location and a message in plain text.</p>
Edit element	Opens the corresponding dialog for editing the selected element.
Show/hide grid	Shows/hides the grid.
Change background color and grid color	Opens the dialog (auf Seite 69) for changing the color of the background and the grid.
Switch to automatic mode	Switches process to automatic mode.
Switch to semi-automatic mode	Switches process to semi-automatic mode.
Switch to manual mode	Switches process to manual mode.
Continue recipe only on selected execution position	Continues a recipe at the selected position.
Continue recipe on all execution positions	Continues a recipe on every available position.
Skip active condition	Skips an active condition.

	Only possible in the manual mode.
<b>Switch recipe to edit mode</b>	Switches from test mode to edit mode.
<b>Release recipe</b>	Releases the recipe. With this a control recipe can be created.

### 7.5.5 Release master recipe

You can release a master recipe by selecting it and clicking button **Release master recipe**.



#### Informazioni

Only master recipes without errors can be released. A released master recipe can no longer be edited.

For each recipe you can create a copy of the released master recipe by clicking on button **Duplicate recipe**. This copy can then be edited.

## 7.6 Control recipe

Control recipes control the progress of a recipe in the Runtime.

### 7.6.1 Tool bar control recipe.

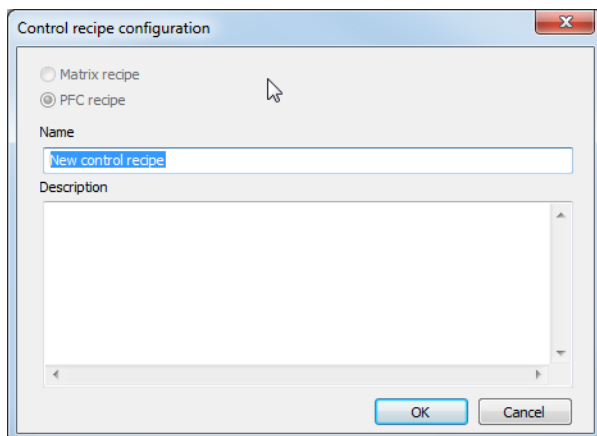


Tag	Type	Description
Start recipe	Command	Starts the recipe process.
Pause recipe	Command	Pauses the recipe.
Resume recipe	Command	Resumes paused recipe.
Hold recipe	Command	Holds recipe.
Restart recipe	Command	Restarts held recipe.
Stop recipe	Command	Stops the recipe.
Abort recipe	Command	Aborts the recipe process.
Pause phase	Command	Pauses the phase.
Resume phase	Command	Resumes the process of a paused phase.
Hold phase	Command	Holds phase.
Restart phase	Command	Restarts held phase.
Edit element	Action	Opens the corresponding dialog for editing the selected element.
Show/hide grid	Action	Shows/hides the grid.
Change background color and grid color	Action	Opens the dialog (auf Seite 69) for changing the color of the background and the grid.
Switch to automatic mode	Command	Switches process to automatic mode.
Switch to semi-automatic mode	Command	Switches process to semi-automatic mode.
Switch to manual mode	Command	Switches process to manual mode.
Continue recipe only on selected execution position	Command	Continues a recipe at the selected position.
Continue recipe on all execution positions	Command	Continues a recipe on every available position.
Skip active condition	Command	Skips an active condition.  Only possible in the manual mode.
Duplicate Recipe	Action	Only active if exactly one recipe is selected. Created a copy of the selected recipe. At the creation of the copy, the version of the recipe saved on the hard disk is used. If the recipe is

		<p>just edited in another computer and the changes have not yet been saved, the changes are not applied. The dialog for the input of a unique name and the description is opened.</p> <p>The copy of the recipe automatically gets the status <i>Prepared</i> and can therefore be edited and started. The REE status (auf Seite 125) of the duplicate is set to <i>automatic</i>.</p> <p>When duplicating a recipe, a CEL entry is created.</p>
--	--	--

## 7.6.2 Create control recipe

You can create control recipes only based on released master recipes. Select the released master recipe in the list of the master recipes, which should serve as basis for your control recipe and click on button **New control recipe...**



Tag	Description
Name	<p>Unique new name for the control recipe. The uniqueness is checked in the entire network. Therefore it can happen that a name is not accepted as another user already used the same name on another computer in the zenon network at the same time.</p> <p>You can change the name afterwards as long as the recipe has status <code>Prepared</code>.</p>
Description	Optional description of the recipe.
OK	Applies configuration and closes the dialog.
Cancel	<i>Discards entries and closes the dialog.</i>

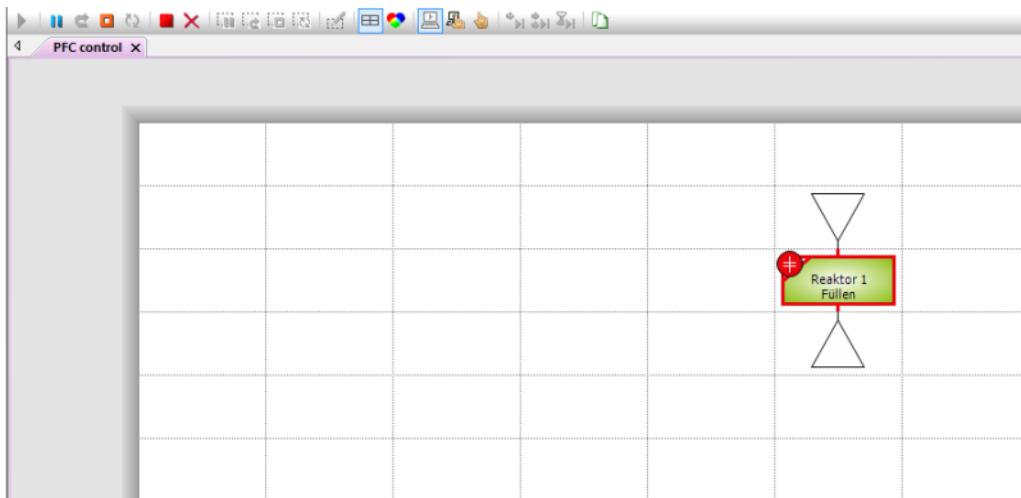
The newly created control recipe is opened on a new tab in the recipe editor if it has been engineered in the screen. The newly created recipe is also displayed in the list of control recipes even if it does not match the set filter criteria. In the network it is only displayed on the computer on which it was created. On other computers the list must be refreshed manually.

### 7.6.3 Execute control recipe

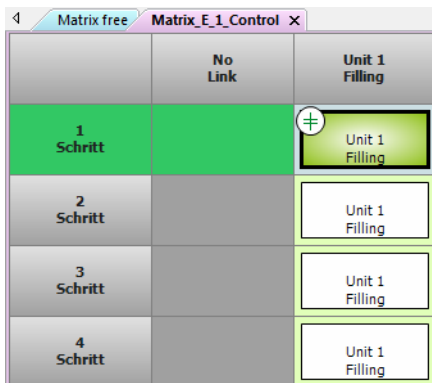
Control recipes can be started:

- ▶ after selecting a control recipe in the list of the control recipes:
  - via click on button `start control recipe`
  - via click on menu item `start` in the context menu
- ▶ via click on symbol `start control recipe` in the tool bar if the control recipe is opened

### PFC control recipe:



### Matrix control recipe:




## CHANGE VALUES

If in the master recipe property `Changeable` in the control recipe was activated, certain values can be adapted in the control recipe as long as the phase is not active yet. In this case a button for synchronization is displayed next to the value. A click on this button takes over the defined value from the master recipe.

### Meaning of the symbols next to the values:



Symbol	Description
= :	Value in the control recipe and in the master recipe match.
<> :	Value in the control recipe and in the master recipe are different.
 :	Click on button to apply the value from the master recipe. It overrides the value in the control recipe. Only active if the values in the control recipe and the master recipe do not match.

## TOOL TIP

A tool tip informs you about the current and historical events of an element.



The following is displayed:

- ▶ Element name
- ▶ general information about status and number of executions
- ▶ Errors
- ▶ Total duration including times in status Paused.
- ▶ Timing

## 7.7 Manage recipes

The entire recipe management is done in the Runtime with the help of one or several screens of type `Batch Control`. Due to suitable filter settings you can achieve already filtered views for master recipes and/or control recipes.

In the screen different control elements (buttons, lists, editors) are available for different tasks. The screen is separated in three main areas which can be used and engineered in part completely independent of each other:

- ▶ Area master recipes (auf Seite 114): Consists of a list and buttons for managing. The area can be used completely independently.
- ▶ Area control recipe (auf Seite 119): Consists of a list and buttons for managing. The area can be used only together with the **list of the master recipes** as a master recipe must be selected first before the appertaining control recipes are displayed in the list.
- ▶ Recipe editors: Depending on the set recipe type the PFC editor (auf Seite 64) or the matrix editor (auf Seite 101) is used. The recipe editor needs either **List master recipe** or both lists for a recipe to be opened in it.

### 7.7.1 Manage master recipes

#### LIST AND LIST FORMATING

List/action	Description
<b>List master recipes</b>	<p>In this list all master recipes can be displayed. The display can be limited by filters to an individual selection.</p> <p>The filtering can be preset in the zenon Editor in the screen switch function (auf Seite 45). Online filtering is also possible. These filters are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p> <p>All commands are also possible in the context menu of the list. The commands for list management can be called from the header of the list. The commands for recipe management can be called at editing one or more recipes.</p> <p>The recipes in the list cannot be edited directly in the list. Renaming, changing the description or changing the recipe status is only possible with the corresponding commands.</p> <p><b>Hint for ideal configuration of the list:</b></p>

The list can be designed diversely concerning content and look:

- ▶ **Content:** The displayed columns (auf Seite 47) can be selected, the column format (auf Seite 48) (column width, alignment, label) can be changed and you can define filters (auf Seite 50). These settings can be edited in Editor and Runtime.
- ▶ **Look:** At the settings of the list in the Editor you can find diverse setting possibilities in areas *Visualizzazione*, *Barra di scorrimento* and *Colori*. With these properties you can even design the list ready for touch operation.

## ACTIONS FOR LIST MANAGEMENT

Action	Description
Column selection master recipe...	<p>Opens a dialog in order to determine which columns should be displayed (auf Seite 47).</p> <p><b>Attention:</b> These changes are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p>
Format columns master recipe...	<p>Opens a dialog to edit the column settings (auf Seite 48).</p> <p><b>Attention:</b> These changes are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p>

## ACTIONS FOR RECIPE MANAGEMENT

Action	Description
New master recipe...	<p>Creates a new and completely empty master recipe in status <code>Editable</code>. The dialog (auf Seite 61) for entering a unique name and a free description is displayed. The uniqueness of the name is also checked in the zenon network.</p> <p>Depending on the license you can also select the recipe type: Matrix recipe (auf Seite 100) or PFC recipe (auf Seite 63). If only one of these recipe types is licensed, the licensed recipe type is fixedly set. The selected recipe type cannot be changed afterwards.</p> <p>When creating a master recipe, a CEL entry is created.</p>
Duplicate master recipe	<p>Only active if exactly one recipe is selected.</p> <p>Created a copy of the selected recipe. At the creation of the copy, the version of the recipe saved on the hard disk is used. If the recipe is just edited in another computer and the changes have not yet been saved, the changes are not applied. The dialog for the input of a unique name and the description is opened.</p> <p>The copy of the recipe automatically receives status <code>Editable</code> and can be edited further.</p> <p>When duplicating a recipe, a CEL entry is created.</p>
Delete master recipe	<p>Deletes the selected recipes irrevocably. If the recipe is opened on another computer for editing, it is automatically closed there.</p> <p>Deleting is only possible if there are no control recipes which are based on the master recipe. First you must delete all control recipes.</p> <p>Recipes which are currently executed in test mode (master recipe status: <code>Test in execution</code>) cannot be deleted. First they must be finished, stopped or canceled.</p> <p>If recipes must not be deleted - e.g. FDA regimented environment - we</p>

	<p>recommend not to engineer this button or to give it an appropriated Livello di autorizzazione:.</p> <p>When deleting a recipe, a CEL entry is created.</p>
<b>Rename master recipe</b>	<p>Only active if exactly one master recipe is selected.</p> <p>The dialog for the input of a unique name and the description is opened. Recipes can only be renamed if they are in status <code>Editable</code>.</p> <p>Also use this function in order to changed the description of the master recipe.</p> <p>When renaming a recipe, a CEL entry is created.</p>
<b>Open master recipe</b>	<p>Opens the selected master recipe in the recipe editor if screen element <code>Recipe editor</code> exists in the screen. Each selected master recipe is opened in a separate tab of the recipe editor.</p>
<b>Release master recipe</b>	<p>Changes the master recipe status of the selected recipes to <code>Released</code>. Only faultless recipes can be released. If error occur during the validation (auf Seite 123), you must first fix them.</p> <p>Only recipes in status <code>Test mode</code> and <code>Editable</code> can be released.</p> <p>Released recipes can no longer be edited. Control recipes can only be created from released recipes.</p> <p>For details about the states see chapter Recipe types and recipe states (auf Seite 58).</p> <p>When releasing a recipe, a CEL entry is created.</p>
<b>Test master recipe</b>	<p>Changes the master recipe status of the selected recipe to <code>Test mode</code>. Only faultless recipes can be switched to test mode. If error occur during the validation (auf Seite 123), you must first fix them.</p> <p>Recipes in the test mode can be executed but no longer reengineered. For details about the states see chapter Recipe types and recipe states (auf Seite 58).</p>
<b>Edit master recipe</b>	<p>Changes the master recipe status of the selected recipes to <code>Editable</code>. In this status recipe can again be edited completely.</p> <p>Only recipes in <code>Test mode</code> can be set back to <code>Editable</code>.</p>
<b>New control recipe...</b>	<p>Opens the dialog (auf Seite 110) for entering a unique name and a description for the control recipe. The uniqueness of the name is also checked in the zenon network. The name must only be unique within the master recipes. Control recipes which are based on other master recipes may have the same name. The uniqueness within module Batch Control is achieved by always referencing the master recipe name and the control recipe name.</p> <p>When creating a control recipe, a CEL entry is created.</p>

## ACTIONS FOR FILLING THE CONTROL RECIPE LIST

As each control recipe can be executed only once, we assume that there are very many control recipes. As during the loading of the list of the control recipes each control recipe is opened on the hard disk, it makes sense to not display all control recipes. Therefore control recipes cannot be opened automatically. They must be called up manually and via filters:

1. Provide the fitting filter options.
2. Select the desired master recipes.
3. Click on button **Display associated control recipes in list**.
4. All control elements complying with the filters and the selection are displayed in the list of the control recipes.

Action/filter	Description
<b>currently executed control recipes</b>	Opens only control recipes which are currently executed. Control recipe status: In execution
<b>prepared control recipes</b>	Opens only control recipes which are prepared for execution. Control recipe status: Prepared
<b>completed control recipes</b>	Opens only control recipes which have already been executed. Control recipe status: Prepared
<b>Display associated control recipes in list</b>	Opens all control recipes which are based on the selected master recipes and which comply with the set filter criteria.

## 7.7.2 Manage control recipes

### LIST AND LIST FORMATING

List/action	Description
List control recipes	<p>In this list all control recipes can be displayed. The display can be limited by filters to an individual selection.</p> <p>Per default the list is empty. For filling the list you must:</p> <ul style="list-style-type: none"> <li>▶ select master recipes</li> <li>▶ set filters currently executed control recipes, prepared control recipes and completed control recipes</li> <li>▶ click button display associated control recipes in list</li> </ul> <p>In addition to the filters mentioned above, you can filter the list itself. The filtering can be preset in the zenon Editor in the screen switch function (auf Seite 45). Online filtering is also possible. These filters are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p> <p>All commands are also possible in the context menu of the list. The commands for list management can be called from the header of the list. The commands for recipe management can be called at editing one or more recipes.</p> <p>The recipes in the list cannot be edited directly in the list. Renaming, changing the description or starting the recipes is only possible with the corresponding commands.</p> <p><b>Hint for ideal configuration of the list</b></p> <p>The list can be designed diversely concerning content and look:</p> <ul style="list-style-type: none"> <li>▶ <b>Content:</b> The displayed columns (auf Seite 47) can be selected, the column format (auf Seite 48) (column width, alignment, label) can be changed and you can define filters (auf Seite 50). These settings can be edited in Editor and Runtime.</li> <li>▶ <b>Look:</b> At the settings of the list in the Editor you can find diverse setting</li> </ul>

	<p>possibilities in areas Visualizzazione, Barra di scorrimento and Colori. With these properties you can even design the list ready for touch operation.</p>
--	---



## ACTIONS FOR LIST MANAGEMENT

Action	Description
Column selection master recipe...	<p>Opens a dialog in order to determine which columns should be displayed (auf Seite 47).</p> <p><b>Attention:</b> These changes are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p>
Format columns master recipe...	<p>Opens a dialog to edit the column settings (auf Seite 48).</p> <p><b>Attention:</b> These changes are discarded when the screen is called up again. A permanent definition is only possible in the zenon Editor.</p>

## ACTIONS FOR RECIPE MANAGEMENT

Action	Description
Duplicate control recipe	<p>Only active if exactly one recipe is selected.</p> <p>Created a copy of the selected recipe. At the creation of the copy, the version of the recipe saved on the hard disk is used. If the recipe is just edited in another computer and the changes have not yet been saved, the changes are not applied. The dialog for the input of a unique name and the description is opened.</p> <p>The copy of the recipe automatically gets the status <code>Prepared</code> and can therefore be edited and started. The REE status (auf Seite 125) of the duplicate is set to <code>automatic</code>.</p> <p>When duplicating a recipe, a CEL entry is created.</p>
Delete control recipe	<p>Deletes the selected recipes irrevocably. If the recipe is opened on another computer for editing, it is automatically closed there.</p> <p>Deleting is only possible if all selected recipes are not executed (control recipe status: <code>In execution</code>). First they must be <code>finished</code>, <code>stopped</code> or <code>canceled</code>.</p> <p>If recipes must not be deleted - e.g. FDA regimented environment - we recommend not to engineer this button or to give it an appropriated <code>Livello di autorizzazione</code>..</p> <p>When deleting a recipe, a CEL entry is created.</p>
Rename control recipe	<p>Only active if exactly one control recipe was selected.</p> <p>The dialog for the input of a unique name and the description is opened.</p> <p>Recipes can only be renamed if they are in status <code>Prepared</code>.</p> <p>Also use this function in order to changed the description of the control recipe.</p>
Open control recipe	<p>Opens the selected control recipe in the recipe editor if screen element</p>

	Recipe editor exists in the screen. Each selected control recipe is opened in a separate tab of the recipe editor.
<b>Start control recipe</b>	Starts the selected control recipe in the defined REE mode. The recipes are executed invisibly at the Server. It is not necessary that the recipe is opened in the recipe editor.

### 7.7.3 Saving on the hard disk and backup scenarios

#### MASTER RECIPES

Each master recipe has a unique ID under which it is saved on the hard disk with file extension `.MR`; e.g. `9.MR`.

Each recipe conforms to one file. The ID of the recipe can be read from the list of the master recipes. For this column `Master recipe ID` must be visible.

The folder for the master recipes is a sub folder of `Cartella Runtime`:  
`\RT\FILES\zenon\system\BatchRecipes`

For the recipe management file `Recipe.unique` is responsible which is located in the same folder. It makes sure that the recipe names are unique.

**Note:** If you delete a recipe manually via the file explorer and therefore outside of the Runtime and the module Batch Control, you must delete file `Recipe.unique` for its content to be correct again. For example if you delete a control recipe manually, you cannot delete the respective master recipe in module Batch Control as the control recipe still exists according to module Batch Control. Only after a reinitialization of file `Recipe.unique` the master recipe can be deleted.

#### BACKING UP MASTER RECIPES

The `.MR` files - and with this all master recipes - can be backed up at any time. For example you can use function `File operations`.

#### RESTORING MASTER RECIPES

The restoring should only be done if absolutely necessary as more current data is overwritten. Proceed as follows:

1. Exit the Runtime.

2. Save all existing master recipes.
3. Rename file `Recipe.unique` or delete it. It automatically re-created at the Runtime start from the `.MR` files.
4. Restore the `.MR` files from an earlier backup.
5. Restart the Runtime.

## CONTROL RECIPE

Each master recipe has a unique ID under which it is saved on the hard disk with file extension `.CR`; e.g. `9.CR`.

Each recipe conforms to one file. The ID of the recipe can be read from the list of the control recipes. For this column `Control recipe ID` must be visible. Control recipes are always based on a master recipe and are therefore always assigned to it. The ID number circles are therefore only unique with regards to the underlying master recipe.

**Example:** The master recipe with ID 9 has the control recipes with IDs 1 and 2. The master recipe with ID 10 has the control recipes with IDs 1 and 2.

Therefore each master recipe has a sub folder in which the control recipes are saved. The name of the folder is always: `<Master recipe ID>.crd`. In our example there is the folder `9.crd` with files `1.CR` and `2.CR` and the folder `10.crd` with files `1.CR` and `2.CR`.

The folder for the command recipes are sub folder of `Cartella Runtime`:

`\RT\FILES\zenon\system\BatchRecipes\`. In this folder the individual control recipe folders have been created. In each control recipe folder there is the file `Recipe.unique`. It makes sure that the recipe names are unique.

## BACKING UP AND RESTORING CONTROL RECIPES

Proceed in the same way as for the master recipes only that you now need to backup all `.CR` files and the appertaining folder structure. At restoring you must delete all `Recipe.unique` files. They are also restored automatically.

## 7.8 Validate recipe

Recipes can be checked for error during the engineering. To validate a recipe click on the corresponding symbol in the tool bar of the recipe editor in the Runtime (green tick - Check recipe for errors). With this the recipe is checked for functionality according to internal rules. The following is especially checked:

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

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










Rules which must be adhered to during the engineering can be found in chapter Engineering rules for recipes (auf Seite 55).

## 7.9 Recipe Execution Engine (REE)

The REE (Recipe Execution Engine) executes recipes in the Runtime. You can start any number of recipes.

### 7.9.1 Symbols and colors in the Runtime

The states during the process of a phase are displayed with the help of different symbols:

Symbol	Meaning
	From the start of the phase; ready for allocation.
	During the waiting for a interlocking condition.
	During the execution of a phase and the waiting for Reaction finished.
	Reaction finished recognized; waiting for Reaction completed.
	Time out for unit allocation expired.
	Time out for interlocking expired.
	Waiting too long for value for writing.
	Time out for Reaction completed expired.
	Time out for Reaction finished.

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

The execution status (auf Seite 125) of a phase is color-coded:

Status	Color
in execution:	green
Timeout:	red border
Held:	gray
Stopped:	yellow
Paused:	orange
Idle, stopped, aborted, resuming, restart:	white
Abort:	red
Completed:	blue

### 7.9.2 The REE states

The following states are possible:

Status	Description
Idle	The REE is in idle state.
In execution	At the start of the control recipe, it changes to status In execution.
Finished	<p>As soon as the execution is finished, the recipe changes to status Finished. In this status execution is not possible.</p> <p>If a control recipe should be executed again after the execution or after the Runtime was exited, its status is automatically set from Finished to Prepared.</p>
Pausing:	The recipe changes to status Paused.
Paused	<p>Within the phase the process stops at:</p> <ul style="list-style-type: none"> <li>▶ Waiting for Finished</li> <li>▶ Waiting for Allocation</li> <li>▶ Waiting for Interlocking condition</li> <li>▶ Waiting for Phase finished</li> <li>▶ Check for parallel execution</li> </ul>
Holding	The object changes to Held and does not carry out any allocations anymore. At the restart the object is restarted and changes to In execution.
Stopped	<p>Within the phase the process stops at:</p> <ul style="list-style-type: none"> <li>▶ Waiting for Finished</li> <li>▶ Waiting for Allocation</li> <li>▶ Waiting for Interlocking condition</li> <li>▶ Waiting for Phase finished</li> <li>▶ Check for parallel execution</li> </ul>
Restarting	Phase is restarting.
Restarted	Phase is completely restarted.
Stopping	Stops the process and changes to Stopped.

Stopped	The object was stopped.
Aborting	Aborts the process and changes to Aborted.
Aborted	Recipe process was aborted.  If a recipe cannot be restarted in the image at the restart, its status automatically changes to Aborted.
Prepared	Prepared for execution. If a control recipe should be executed again after the execution or after the Runtime was exited, its status is automatically set from Finished to Prepared.

## EXECUTION POSITION AND JUMP TARGETS

Status	Description
Resumption	If an object is paused and after the object an execution position is located, Resume has the same effect as Next step. This also includes jumps.  At a phase command the command only effects a 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 phase command only the jump in the area of the phase is deleted.

### 7.9.3 Mode and mode change

The REE can run in three modes:

- ▶ **Automatic:** The recipe runs entirely automatically.
- ▶ **Semi-automatic:** The recipe is executed manually. Conditions cannot be jumped.
- ▶ **Manual:** Each step in the recipe is executed manually. Conditions on which it is waited can be jumped (forced).

To execute a recipe manually or semi-automatically, you can use the operation types (auf Seite 128) **Step-by-step execution of the recipe** and **Jump**.

To react on serious events, you can change the mode during the running process via reaction type (auf Seite 23) influencing the recipe.

## 7.9.4 Step-by-step execution of a recipe and jumps in the recipe

### STEP-BY-STEP EXECUTION OF A RECIPE

A recipe can be executed step-by-step if:

- ▶ the recipe is in mode semi-automatic or manual  
and
- ▶ the status of the recipe is `In execution`.

For the step-by-step execution the execution is held as soon as an element is finished with its execution. The holding is done via command `Pause` to the concerned execution path. As soon as all active elements in this path reached status `Paused`, the execution position is marked by an orange arrow.

The execution is resumed with:

- ▶ a selective step: selection of the corresponding arrow (green)
- ▶ a global step: all positions with arrows for possible resuming are started

### COMMANDS

#### GLOBAL COMMANDS

For global commands all execution positions are deleted as the execution cannot be resumed from there.

**Exceptions:** `Pause` and `Resume`. Here the execution positions remain.

#### PHASE COMMANDS

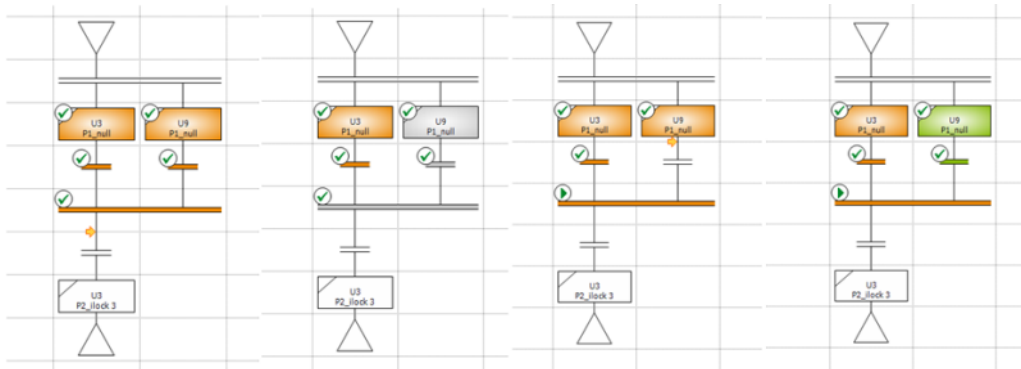
**Hold for a phase** The execution position is deleted from the execution path of the phase.

**Resume:** If an execution position exists, a selective next step is executed in this execution path.



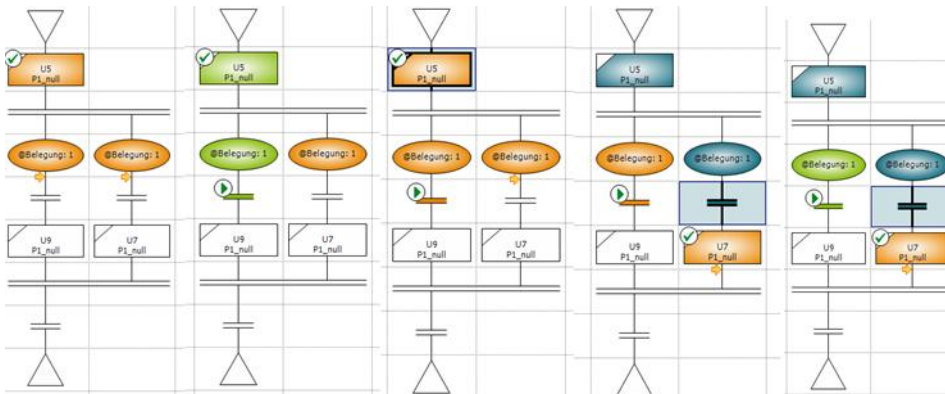
## SPECIAL CASES

### PHASE COMMAND HOLD AND RESTART IN A PARALLEL BRANCH FOR ACTIVE END PARALLEL BRANCH



If you hold in this example (images from left to right) a parallel branch and restart it, then you will reach after a step-by-step execution the already paused **end parallel branch**. To resume the execution from here, the left phase must receive command **Resume**.

### ONLY ONE PATH IN A PARALLEL BRANCH WITH AN ACTIVE PHASE BEFORE STEP-BY-STEP EXECUTION



If in a parallel branch with a phase before only one path executed completely and in one of the other parallel branches waited (phase before is **In execution**), in the parallel branch no execution position is displayed. To get them in the other path, the phase must be paused with a phase command. After that it is possible to execute the path completely.

If then however the parallel path with the execution position is still executed, the phase before the parallel branch is deactivated. With this the left path remains in **Paused** and without execution position. As there is not active phase, the execution can only be resumed with a global **Pause** and **Resume**.

## JUMP

Jump means to move from one position to another, distant position during execution in order to continue the execution there.

To jump:

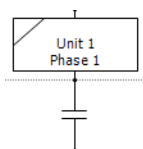
1. select an execution position with the mouse cursor
2. move it to one of the offered targets
3. execute the next step

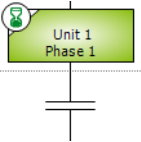
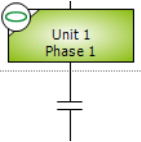
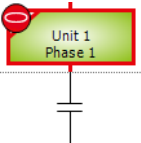
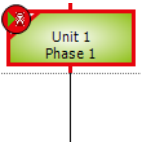
All active elements in front of the source pointer are deactivated and the object after the target is activated.

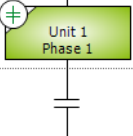
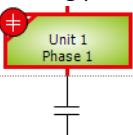
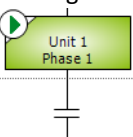
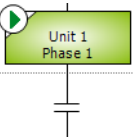
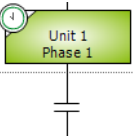
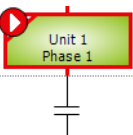
If a jump is registered for which source and target are analogously the same (jumps over lines, jump targets or end branch objects), this jump is ignored and a simple step is executed.

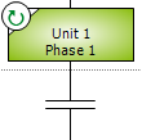
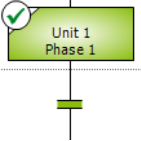
### 7.9.5 Process of a phase in detail

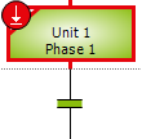
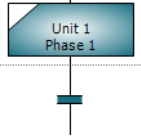
A phase is always processed sequentially after the same pattern. To break down the exact process, you also need a following condition. For this display we use a transition as following condition. We give the phase the name **Phase 1**. You can find additional special process behavior with following condition in chapter Following condition (auf Seite 134).



Phase	Transition	Event
<p>Phase is activated</p> 		
<p>All variables of the phase are registered at the drivers.</p>		
<p>All events of the phase are activated.</p>		Phase started
<p>Unit allocation is started and Waiting period unit allocation is started.</p> 		<p>If the unit allocation was not successful in the first try: Unit allocation not possible</p>
<p>Optional: Unit allocation possible within the waiting period.</p> 		Waiting period unit allocation exceeded
<p>Check is started whether phase isn't already executed. This can happen if the phase is already active in a parallel branch or if the unit allocation was skipped manually and the phase is already executed in another recipe.</p> 		Phase started multiple times
<p>Check of the input interlocking is started and Waiting period input interlocking is started.</p>		<p>If the checking of the input interlocking was not successful in the first try: Input interlocking blocked</p>

		
<p>Optional: Condition of the input interlocking not TRUE within the waiting period.</p> 		Waiting period input interlocking exceeded
<p>Writing the command tag</p> 		When all command tags were written: Finished writing command tags
<p>Checking of the phase-done condition is started and time for Minimum execution duration and Maximum execution duration is started.</p> 		
<p>Optional: Phase-don condition fulfilled but Minimum execution duration not reached.</p> 		
<p>Optional: Condition of the Phase-done condition not TRUE within the Maximum execution duration.</p> 		Waiting period phase-done condition exceeded

<p>Optional: Waiting for restart of the whole execution. If the execution is still restarting (for other phases in the recipe the restart condition is not yet fulfilled), it is waited here. This guarantees that the following element is activated after the recipe changes to <b>In execution</b>.</p> 		
<p>Phase-done condition is <b>TRUE</b> and minimum execution duration is reached or exceeded.</p>  <p>Waiting period following condition is started.</p>	Transition is activated	Phase finished
	All variables of the transition are registered at the drivers.	Waiting period following condition exceeded

<p>Optional: Following condition not within waiting period TRUE</p> 	<p>The transition condition is checked.</p>	
<p>The next phase is activated. The following condition can be composed from several objects (e.g. transition + unit allocation). No till the next phase is reached (or the end of the recipe), the following condition counts as fulfilled.</p>		
<p>Phase is informed that the following condition is fulfilled.</p> 	<p>Transition condition is TRUE.</p>	<p>Phase completed</p>
<p>All events of the phase are deactivated.</p>		
<p>All variables of the phase are signed off from the drivers.</p>	<p>All variables of the transitions are signed off from the drivers.</p>	
<p>The phase is deactivated.</p>	<p>The transition is deactivated.</p>	

## RULE FOR VALUES OF TRANSITIONS

If a transition has value `TRUE` for the phase-done condition during the waiting period, it is marked as *finished*. If its value should later change to `FALSE`, the execution of the recipe is not influenced.

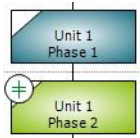
## Following condition

A phase is active as long as the following condition is fulfilled. Only when the following condition is fulfilled, the phase is completed. The event `Phase completed` is triggered and the phase is deactivated. Before it is deactivated, the event reactions are executed. See also Process of a phase in detail (auf Seite 130)

The following condition can be very different. Here some examples:

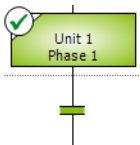
► Phase 1 followed by a phase 2:

As soon as the phase done condition (and optional the Minimum execution duration) is fulfilled for phase 1, phase 1 is completed and phase 2 is activated.



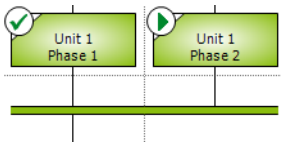
► Phase 1 followed by a transition:

Only when the transition condition is fulfilled, phase 1 is completed.



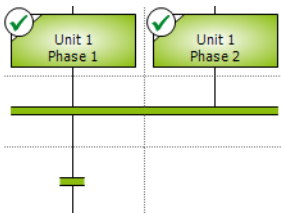
► Phase 1 and phase 2 parallel followed by an end parallel branch:

Not before the phase done condition (and optional the minimum execution duration) is fulfilled for both phases, both phases are completed.



► Phase 1 and phase 2 parallel followed by an end parallel branch followed by a transition:

Only when the transition condition is fulfilled, both phases are completed.



## 7.9.6 Exit and restart Runtime

At closing the Runtime an image file of the running recipe is created. It contains the images of the REEs, the order of the allocation and all reactions which are needed for the restart.

## ALLOCATIONS

After the restart the allocations match the state before the finishing. It is saved who allocated a unit and who forced an allocation in which order. If a recipe cannot be restarted (e.g. because of failed validation), the allocations for this recipe are removed.

## REACTIONS

Reactions which were triggered by the process are also incorporated in the image if they are active. They are then executed after the REE is restarted. This guarantees that the reaction is always executed as a whole either before the image file is created or after the restart.

The `Exit Runtime` reaction is always executed and can never be incorporated in the image.

## SYNCHRONOUS WRITING

The REE manages the confirmation for all variables whose write set value should be executed synchronously. The time out for this is defined by the time within which the Runtime must be closed. For each write acknowledgment the time out is restarted. A time out is written in the log file.

Variables which don't access a driver are always written without an acknowledgment even if an acknowledgment is requested. Internal drivers do not support acknowledgments.

## ALLOCATE TAG

As during the start of the Runtime all drivers are also started, it is possible that they do not provide valid values if they are needed at the restart. During the restart it is not waited for the value update. This does not ensure that the value is written as expected. If no value is available, the alternate value is used.

It is waited for the values of the internal driver if they are available within 2 cycles.

### 7.9.7 Restart phase

Phases can be restarted. At this linked active objects are deactivated.

## DEACTIVATION

At the restart of a phase, all active objects are deactivated in principle, however only if they are linked. Isolated active objects are not deactivated.



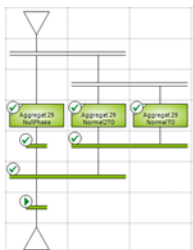
## SEQUENCE OF DEACTIVATION

The deactivation starts from the object which has been active the longest. After the restart this object is also going to be activated. If this object is deactivated, it also deactivates all branches to which it can establish a connection via an active object.

## ISOLATED BRANCH

A branch is isolated if it is not connected to another branch via an active object. The deactivation only takes place between connected branches. As long as an object does not have status `Completed`, there is not active connection to the following object. If no phase exists in an isolated branch, at the restart the oldest object is activated; e.g. a transition.

## EXAMPLE



If in this process `NormalTO` is restarted, `end parallel branch` and the `transitions` are also deactivated. They are in outside branches seen from the `NormalTO`.

This means for `end parallel branch`: As long as not all paths are active at `end parallel branch`, there is no active connection to the following objects. For `begin parallel branch` the same is true.

As long as the transition from `begin parallel branch` is active, `begin parallel branch` and both `begin branch` are active. A restart will also restart the inner branches. A restart of the inner branches deactivates them. The outmost transition is restarted.

## RESTART OF SELECTED PHASES

One or more phases can be selected and restarted. For the selection of several phases, they must be in separate branches.

The oldest object is restarted in the selected branch. With this all active objects in the connected branches are deactivated.

## GLOBAL RESTART

The global restart carries out a restart for all phases. The restart is done for the oldest active object and with this all connected, active objects are deactivated. All remaining active objects are in an isolated branch. Here also the oldest object is restarted until all active objects were dealt with.

## 8. Behavior in the network

The module **Batch Control** is fully capable of using a network in terms of Client/Server technology. This means that Batch recipes can be created, duplicated, edited, deleted, etc. on a Client. The whole recipe management remains always on the server. Likewise the whole process control such as **start** recipe, **pause** recipe, **stop** recipe, etc. can be done from the Client. Also mode changes and manual operations such as **jump** are possible.



### Attenzione

*Module Batch Control does not support redundancy. There is no synchronization between Standby-Server. When the Server breaks down, the executed Batch recipes are not continued seamlessly on the Standby!*

For using Batch Control in a network the following is true:

## ALLOCATION

- The forcing of allocations can be carried out by Server and Client.

## FUNCTIONS

Functions are always carried out at the Server.

## PHASES

- Editing phases in the master recipe:
  - Edit mode: Changes are done locally at the Client.  
If during the editing the recipe is saved on another computer in the network, the current configuration is lost. An appropriate message is displayed and the editing dialog is closed. The new data from the server are displayed.

- Test mode: Changes a done at the Server.
- ▶ Control recipe: Changes a done at the Server.
- ▶ If a recipe is saved in the network, all Clients using this recipe are updated.
- ▶ If a recipe is deleted on a computer, a message is displayed on all computer on which the recipe is opened that the recipe has been deleted.

## MODE

- ▶ The mode (automatic, semi-automatic, manual) can be switched by the Server and the Client.
- ▶ Jumps in the recipe and step-by-step progress of a recipe can be done from Server and Client.

## RECIPES

- ▶ Recipes can be started and controlled from Server and Client.

## WEB CLIENT

With a standard web client it is not possible to change the size of a sheet. Changes for color and grid are possible. You also cannot create new recipes. In the tool bar all symbol which are not allowed are deactivated. You cannot select these objects.

The web client PRO is not affected by these restrictions.

# 9. Formula editor

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

### Editor:

- ▶ Phases
- ▶ Interlocking conditions
- ▶ all conditions for transitions
- ▶ Phase done condition

### Runtime:

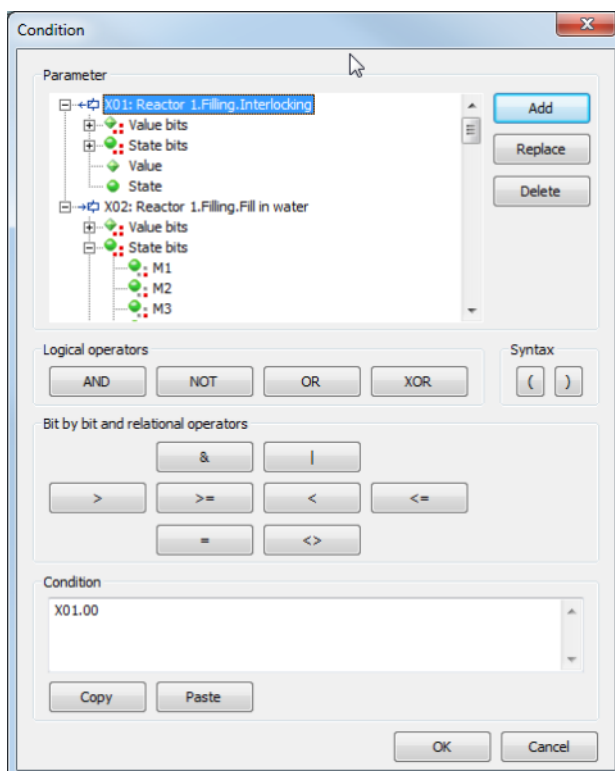
- ▶ Phase done condition and interlocking when editing a phase
- ▶ Editing transitions
- 

**Note:** If the phase referenced in the formula is removed and a new phase is added, the operands are assigned newly.

## ENTER FORMULA

The following input is accepted:

- ▶ Constant as decimal number
- ▶ Hexadecimal number if it is preceded by an  $\times$
- ▶ 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



Tag	Description
<b>Tag list</b>	<p>List of the tags which can be used for the formula.</p> <p>Each entry contains of:</p> <ul style="list-style-type: none"> <li>▶ a basis node for the label</li> <li>▶ a value</li> <li>▶ a status</li> <li>▶ the bits for value and status</li> </ul> <p>A symbol at the first node displays whether it is a command or return tag.</p> <p>The short identifier at the beginning of the name is used for the formula.</p>
<b>Add</b>	<p>Opens the dialog for adding a parameter. For this, the following applies:</p> <ul style="list-style-type: none"> <li>▶ The following can be added: numeric and binary tags and tags for time duration.</li> <li>▶ For conditions of the phase only the tags created for it can be added.</li> <li>▶ Tags can be added multiple times.</li> </ul>
<b>Replace</b>	Makes it possible to replace a tag. The short identifier remains the same at replacing.
<b>Remove</b>	<p>Removes the highlighted tag. For a tag to be deleted:</p> <ul style="list-style-type: none"> <li>▶ the formula must be correct</li> <li>▶ the selected tag must not be used in the formula</li> </ul>
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 comparison operators	
&	And
	Or
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less then or equal
=	Equal
<>	Less than or greater than
<b>Condition</b>	Configuration and display of the formula.
<b>Copy</b>	<p>Copies the whole formula:</p> <ul style="list-style-type: none"> <li>▶ All configured tags from the tag tree</li> <li>▶ Formula from the field</li> </ul>
<b>Paste</b>	<p>Pastes a formula from the clipboard. At this all already configured elements are deleted and replaced by the copied formula.</p> <p>When copying formulas between phases, it is tried 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.</p>
<b>OK</b>	<p>Applies formula and closes the dialog.</p> <p>For this the formula must be correct.</p>
<b>Cancel</b>	Discards all changes and closes the dialog.



### Informazioni

*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:

Tag	Description
Value bits	There are 32 value bits (0 - 31) 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.
Status bits	Provides the most commonly used status bits. You can find the exact definition of the status bits in chapter List status bits.
Value and status	<p>All values (value bits and status bits) in the formula are considered as binary value and can be linked with logical operators such as AND or OR.</p> <p>The total value and the total status are the exception. In order to get a Boolean result this total value has to be ORed with a constant <code>bitwise</code>. For this the operator <code>&amp;</code> is used.</p> <p>The result 0 (false) of the bit by bit ORing results in the binary value 0 (false) otherwise 1 (true).</p> <p>Example: see chapter Example bit by bit ORing</p>



### Info

*The status bits NORM and N\_NORM are only available in the formula editor and cannot be engineered via the status.*



### Informazioni

Formulas with binary x values and bit-by-bit link can be used with a maximum of 2 binary values. If more values are necessary the link must be done without binary x values.

#### Example:

`X01.Value & X02.Value` -> works

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

But:

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



## 9.1 List status bits

Bit numero	Denominazione breve	Denominazione lunga	zenon Logic indicatore
0	M1	Stato utente 1	_VSB_ST_M1
1	M2	Stato utente 2	_VSB_ST_M2
2	M3	Stato utente 3	_VSB_ST_M3
3	M4	Stato utente 4	_VSB_ST_M4
4	M5	Stato utente 5	_VSB_ST_M5
5	M6	Stato utente 6	_VSB_ST_M6
6	M7	Stato utente 7	_VSB_ST_M7
7	M8	Stato utente 8	_VSB_ST_M8
8	NET_SEL	Select in rete	_VSB_SELEC
9	REVISION	Revisione	_VSB_REV
10	PROGRESS	In uso	_VSB_DIRECT
11	TIMEOUT	Runtime exceeded	_VSB_RTE
12	MAN_VAL	Valore manuale	_VSB_MVALUE
13	M14	Stato utente 14	_VSB_ST_14
14	M15	Stato utente 15	_VSB_ST_15
15	M16	Stato utente 16	_VSB_ST_16
16	GI	Interrogazione generale	_VSB_GR
17	SPONT	Spontaneo	_VSB_SPONT
18	INVALID	Non valido	_VSB_I_BIT
19	T_CHG_A	Cambio orario estivo/invernale	_VSB_SUWI
20	OFF	Spenta	_VSB_N_UPD
21	T_EXTERN	Tempo reale esterno	_VSB_RT_E
22	T_INTERN	Tempo reale interno	_VSB_RT_I
23	N_SORTAB	Non filtrati	_VSB_NSORT

24	FM_TR	Transformer	_VSB_DM_TR
25	RM_TR	Transformer anomaly	_VSB_RM_TR
26	INFO	Info della variabile	_VSB_INFO
27	ALT_VAL	Valore di riserva:  Se non è stato trasmesso ancora nessun valore, viene usato il valore di sostituzione definito; in caso contrario l'ultimo valore valido.	_VSB_AVALUE
28	RES28	Riservato per uso interno (lampeggio allarmi)	_VSB_RES28
29	N_UPDATE	Non attualizzato	_VSB_ACTUAL
30	T_STD	Ora solare	_VSB_WINTER
31	RES31	Riservato per uso interno (lampeggio allarmi)	_VSB_RES31
32	COT0	Causa trasmissione Bit 1	_VSB_TCB0
33	COT1	Causa trasmissione Bit 2	_VSB_TCB1
34	COT2	Causa trasmissione Bit 3	_VSB_TCB2
35	COT3	Causa trasmissione Bit 4	_VSB_TCB3
36	COT4	Causa trasmissione Bit 5	_VSB_TCB4
37	COT5	Causa trasmissione Bit 6	_VSB_TCB5
38	N_CONF	Conferma negativa del select dell'apparecchio (IEC 60870 [P/N])	_VSB_PN_BIT
39	TEST	Test-Bit (IEC 60870 [T])	_VSB_T_BIT
40	WR_ACK	Conferma scrittura	_VSB_WR_ACK
41	WR_SUC	Scrittura avvenuta con successo	_VSB_WR_SUC
42	NORM	Stato normale	_VSB_NORM
43	N_NORM	Normal deviation	_VSB_ABNORM
44	BL_870	IEC 60870 Status: blocked	_VSB_BL_BIT
45	SB_870	IEC 60870 Status: substituted	_VSB_SP_BIT

46	NT_870	IEC 60870 Status: not topical	_VSB_NT_BIT
47	OV_870	IEC 60870 Status: overflow	_VSB_OV_BIT
48	SE_870	IEC 60870 Status: select	_VSB_SE_BIT
49	T_INVAL	Timestamp non valido	non definito
50	CB_TRIP	Segnalazione di Switch riconosciuta	non definito
51	CB_TR_I	Segnalazione di Switch non attiva	non definito
52	RES52	riservato	non definito
53	RES53	riservato	non definito
54	RES54	riservato	non definito
55	RES55	riservato	non definito
56	RES56	riservato	non definito
57	RES57	riservato	non definito
58	RES58	riservato	non definito
59	RES59	riservato	non definito
60	RES60	riservato	non definito
61	RES61	riservato	non definito
62	RES62	riservato	non definito
63	RES63	riservato	non definito



### Informazioni

*In formule sono disponibili tutti i bit di stato. Per altri usi la disponibilità può essere ridotta.*

*Dettagli per la modifica degli stati li potete trovare nel capitolo Modifica stati.*

## 9.2 Logical operators

Collegamenti logici: Le variabili vengono controllate solo sulla base del valore logico '0'; nel caso in cui il valore non sia pari a '0', viene definito con '1'.

A differenza del Bitformel (??) l'area tecnica può essere modificata di un fattore di estensione -> diverso da 0 o 1.

Operatore	Significato
AND	'AND' logico
NOT	Negazione
OR	'OR' logico
XOR	'ESCLUSIVO OR' logico

Gli operatori hanno le seguenti priorità nel calcolo della formula:

Priorità	Operatore
1	& (operatore per Formula bit)
2	NOT
3	AND
4	XOR/OR



### Info

*Possono essere collegate solo fino a 99 variabili in una formula. Da X01 fino a X99.*



### Info

*I bit di stato NORM e N\_NORM sono disponibili solo in questa sede, nell'Editor delle formule, e non sono configurabili mediante lo stato.*

## 9.3 Bit formula

Le formule bit possiedono solamente uno stato logico low e high. A differenza di quanto accade per le formule logiche, il valore raw è predefinito (0,1).

Operatore	Descrizione
&	AND
	OR

### 9.3.1 Example bit by bit OR function

Volete sapere se è settato uno stato user bits 1-8 (M1 ... M8) della variabile X01.

#### FORMULA NORMALE:

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

Questa richiesta può essere facilitata di molto attraverso il collegamento ad "o" a bit dello stato generale.

#### COLLEGAMENTO A "O"

`X01.Status & 0xFF`

La costante può essere indicata in forma esadecimale, come descritto sopra.

0xFF corrisponde al valore decimale 256; sono i primi 8 bit di stato (binario 11111111). Se uno di questi bit è a 1, il risultato del collegamento con operazione OR a bit è 1 (True), altrimenti 0 (False).

Se si vuole richiedere, ad esempio, tutti gli stati bit user fatta eccezione per lo status bit user M7, l'espressione binaria corrispondente sarebbe: 10111111. Bit 7 non è interessante, per questo è settato su 0. Ciò corrisponde a 0xBF esadecimale. L'espressione per la formula sarebbe allora: `X01.Status & 0xBF`.

Invece di collegare con operazione OR a bit con una costante, si può anche confrontare direttamente il valore con un numero decimale. Se il confronto dà un esito sbagliato, ne deriva il valore binario 0 (False), altrimenti 1 (True).

#### Beispiel:

Volete interrogare se un valore corrisponde alla costante 202. La formula è la seguente: La formula corrispondente è la seguente:

`X01.Wert = 202`

Se il valore corrisponde alla costante 202, il risultato di questo confronto è 1 (True), altrimenti 0 (False).

**Indicazione:** il collegamento O in bit funziona con il segno O (1) in modo analogo al questo esempio.


## 9.4 Comparison operators

Gli operatori a confronto servono al confronto diretto di due valori numerici. Il risultato di questo confronto viene espresso in forma di uno stato binario. „0“ se la condizione non è soddisfatta e „1“ se invece lo è

Operatore	Descrizione
<	Minore:
>	Maggiore:
<=	minore uguale
>=	maggiore uguale
=	è uguale
<>	disuguale

A sinistra e a destra di un operatore a confronto deve stare un valore (generale) o uno stato (generale); bit singoli non possono essere utilizzati con questi operatori a confronto.

A destra dell'operatore a confronto ci può essere anche una costante. (le costanti possono essere solo numeri interi; un confronto con numeri con cifre decimali non è possibile)

Nell'elemento combinato queste costanti vengono inserite come valori esadecimali o valori decimali. I numeri esadecimali vengono trasformati automaticamente in numeri decimali cliccando su  (per es. 0x64 corrisponde al valore numerico 100).

**Esempio**

*X01.valore >= X02.valore*

*Il risultato è „1“ se il valore di X01 è più grande o uguale al valore X02*

*X01.valore = 0x64*

*Il risultato è „1“ se il valore di X01 corrisponde esattamente al valore numerico 100 (= Hex 0x64)*

*(X01.Wert = 0x64) OR (X01.Wert = 0x65)*

*Il risultato è „1“ se il valore di X01 corrisponde esattamente al valore numerico 100 o 101 (= Hex 0x64 und Hex 0x65)*

## 9.5 Examples for formulas

### SEMPLICE COLLEGAMENTO AND FRA DUE VALORI BIT.

**Esempio**

*Formula: X01.03 AND X02.03*

Questa formula dà come risultato “VERO” quando il **Bit 3** della variabile 1 e il **Bit 3** della variabile 2 hanno entrambi il valore 1.

### CONFRONTA IL VALORE O LO STATO DI UNA VARIABILE ANALOGICAMENTE

**Esempio**

*(X01.Wert > X02.Wert)*

## CONFRONTA FRA DI LORO COMPARAZIONI ANALOGICHE SU BASE LOGICA.



### Esempio

$(X01.Wert > X02.Wert) \text{ AND } (X01.Wert = X02.Wert)$

## CONFRONTI DI BIT VALORE E BIT DI STATO



### Esempio

$(X01.Wert > X02.Wert) \text{ AND } (X01.Wert = X02.Wert) \text{ OR } (X01.03 = X02.03)$


## CONFRONTA UN VALORE CON UNO DECIMALE O ESADECIMALE.



### Esempio

Formula:  $(X01.Wert = 111)$

Formula:  $(X01.Wert = 0x6F)$

Se si usa un valore esadecimale, esso viene trasformato più tardi nella modalità di scrittura decimale cliccando su . Se si è indicato e confermato un valore decimale, alla riapertura del valore esso continua ad essere visualizzato in forma decimale.



### Info

*Non è possibile nell'indicazione di numeri usare una virgola o un punto.*

## 10. Troubleshooting

Warnings and error messages are written in log files and can be analyzed with the Diagnosis Viewer. For this you must activate module Batch Control in the filter settings.



## SYSTEM DRIVER VARIABLE

Batch Control provides the system driver `SYSDRV` with information via system driver variables. For information about there messages see manual `SYSDRV` (`Main.chm: /Sysdrv.chm: /Sysdrv.htm`) in chapter `Topic - Batch Control` (`sysdrv.chm: /34270.htm`).

## POP-UP MESSAGES

Message	Description
The name "name" is invalid or is already used.	The name either contains invalid characters or is already used.  The name must not be empty, must not contain a dot and must not contain only of spaces.
Duplicating '<MR>' @rejected by the server.	The recipe selected for duplication could not be duplicated at the Server. Maybe it no longer exists.
Do you really want to delete the selected master recipes?	Confirmation message before deleting the master recipes irrevocably.
The diagram was changed. Do you want to save it?	Confirmation message for saving before closing an unsaved diagram.