

LISTEN.
THINK.
SOLVE.®

PharmaSuite®



EQUIPMENT PHASES

RELEASE 8.4

USER MANUAL

PUBLICATION PSEQ-UM004F-EN-E-DECEMBER-2017
Supersedes publication PSEQ-UM004E-EN-E



Allen-Bradley • Rockwell Software

Rockwell
Automation

Contact Rockwell See contact information provided in your maintenance contract.

Copyright Notice © 2017 Rockwell Automation Technologies, Inc. All rights reserved.
This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation Technologies, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation Technologies, Inc. is strictly prohibited. Please refer to the license agreement for details.

Trademark Notices FactoryTalk, PharmaSuite, Rockwell Automation, Rockwell Software, and the Rockwell Software logo are registered trademarks of Rockwell Automation, Inc.

The following logos and products are trademarks of Rockwell Automation, Inc.:

FactoryTalk Shop Operations Server, FactoryTalk ProductionCentre, FactoryTalk Administration Console, FactoryTalk Automation Platform, and FactoryTalk Security.

Operational Data Store, ODS, Plant Operations, Process Designer, Shop Operations, Rockwell Software CPGSuite, and Rockwell Software AutoSuite.

Other Trademarks ActiveX, Microsoft, Microsoft Access, SQL Server, Visual Basic, Visual C++, Visual SourceSafe, Windows, Windows 7 Professional, Windows Server 2008, Windows Server 2012, and Windows Server 2016 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

ControlNet is a registered trademark of ControlNet International.

DeviceNet is a trademark of the Open DeviceNet Vendor Association, Inc. (ODVA).

Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.

OLE for Process Control (OPC) is a registered trademark of the OPC Foundation.

Oracle, SQL*Net, and SQL*Plus are registered trademarks of Oracle Corporation.

All other trademarks are the property of their respective holders and are hereby acknowledged.

Warranty This product is warranted in accordance with the product license. The product's performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product's implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at any time without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Chapter 1	Equipment Phases	1
	Typographical Conventions	2
Chapter 2	Equipment Tracking.....	3
	Binding of Equipment	3
	Grouping of Equipment	4
	Creating an Equipment Entity Group	5
	Identifying an Equipment Entity Group for Use	8
	Re-identifying an Already Bound Equipment Entity Group	8
	Triggering a Graph Transition on an Equipment Entity Group	10
	Separating an Equipment Entity Group.....	11
Chapter 3	Identify Equipment.....	19
	Execution	20
	Phase Design	23
	Process Parameters	24
	Output Variables	29
Chapter 4	Change Equipment Status.....	33
	Execution	33
	Phase Design	34
	Process Parameters	35
	Output Variables	38
Chapter 5	Trigger Graph Transition.....	41
	Execution	41
	Phase Design	43
	Process Parameters	44

	Output Variables	47
Chapter 6	Separate Equipment	51
	Execution	51
	Phase Design	54
	Process Parameters	55
	Output Variables	58
Chapter 7	Unbind Equipment	61
	Execution	61
	Phase Design	63
	Process Parameters	63
	Output Variables	66
Chapter 8	Get OPC Values	69
	Execution	70
	Phase Design	72
	Process Parameters	72
	Output Variables	81
Chapter 9	Set OPC Values	87
	Execution	88
	Phase Design	90
	Process Parameters	90
	Output Variables	96
Chapter 10	Monitor Numeric Value	101
	Execution	101
	Phase Design	103
	Process Parameters	104
	Output Variables	108
Chapter 11	Get Alarms	111
	Execution	111
	Phase Design	113

Process Parameters	114
Output Variables	117
Chapter 12 Show Historical Data Chart	121
Execution	121
Phase Design	123
Process Parameters	123
Output Variables	128
Index	131

-
-
- Rockwell Software PharmaSuite® BB - User Manual Equipment Phases
-
-

Figure 1: Structure of an equipment entity group	4
Figure 2: Sample group with assembly operation	5
Figure 3: Sample group with re-use status operation	10
Figure 4: Full separation with release of binding	12
Figure 5: Full separation with retained grouping	13
Figure 6: Partial separation with dissolved grouping	14
Figure 7: Partial separation with retained grouping	15
Figure 8: Sample group with operation for partial separation	16
Figure 9: Identify equipment during execution	21
Figure 10: Identify equipment group during execution	21
Figure 11: Identify equipment group for grouping during execution	22
Figure 12: User-triggered exceptions of Identify equipment	22
Figure 13: Identify equipment after phase completion	22
Figure 14: Identify equipment group after phase completion	22
Figure 15: Identify equipment group for grouping after phase completion	23
Figure 16: Identify equipment in the Navigator	23
Figure 17: Identify equipment group in the Navigator	23
Figure 18: Change equipment status during execution	34
Figure 19: User-triggered exception of Change equipment status	34
Figure 20: Change equipment status after phase completion	34
Figure 21: Change equipment status in the Navigator	34
Figure 22: Change Action Selection editor	36
Figure 23: Trigger graph transition during execution	42
Figure 24: Trigger graph transition of group during execution	42
Figure 25: Trigger graph transition after phase completion	42
Figure 26: Trigger graph transition of group after phase completion	43
Figure 27: Trigger graph transition in the Navigator	43

Figure 28: Trigger graph transition of group in the Navigator	43
Figure 29: Trigger Selection editor	46
Figure 30: Separate equipment (single entity) during execution.....	52
Figure 31: Separate equipment (entity group) during execution.....	52
Figure 32: Separate equipment (all entities) during execution.....	53
Figure 33: User-triggered exception of Separate equipment	53
Figure 34: Separate equipment (single entity) after phase completion.....	53
Figure 35: Separate equipment (entity group) after phase completion.....	53
Figure 36: Separate equipment (all entities) after phase completion.....	54
Figure 37: Separate equipment (single entity) in the Navigator	54
Figure 38: Separate equipment (entity group) in the Navigator.....	54
Figure 39: Separate equipment (all entities) in the Navigator	54
Figure 40: Unbind equipment during execution	62
Figure 41: Unbind equipment group during execution	62
Figure 42: Unbind equipment after phase completion	62
Figure 43: Unbind equipment group after phase completion	62
Figure 44: Unbind equipment in the Navigator	62
Figure 45: Unbind equipment group in the Navigator	62
Figure 46: Get OPC values during execution	70
Figure 47: User-triggered exceptions of Get OPC values	71
Figure 48: Get OPC values after phase completion	71
Figure 49: Get OPC values in the Navigator	71
Figure 50: Property Selection editor for Automation properties (Boolean)	75
Figure 51: Property Selection editor for Automation properties (Numeric)	77
Figure 52: Property Selection editor for Automation properties (String)	80
Figure 53: Set OPC values during execution.....	88
Figure 54: User-triggered exceptions of Set OPC values	89
Figure 55: Set OPC values after phase completion.....	89
Figure 56: Set OPC values in the Navigator	89
Figure 57: Property Selection editor for Automation properties (Boolean)	94
Figure 58: Property Selection editor for Automation properties (Numeric)	95
Figure 59: Property Selection editor for Automation properties (String)	96
Figure 60: Monitor numeric value during execution	102

Figure 61: User-triggered exception of Monitor numeric value.....	102
Figure 62: Monitor numeric value after phase completion	103
Figure 63: Monitor numeric value in the Navigator	103
Figure 64: Property Selection editor for Automation properties (Numeric)	106
Figure 65: Get alarms during execution	112
Figure 66: Get alarms after phase completion	113
Figure 67: Get alarms in the Navigator	113
Figure 68: Property Selection editor for Automation properties (FlexibleTagDefinition).....	115
Figure 69: Show historical data chart during execution	122
Figure 70: User-triggered exception of Show historical data chart	122
Figure 71: Show historical data chart after phase completion	122
Figure 72: Show historical data chart in the Navigator	122
Figure 73: Property Selection editor for Historian properties	126

Equipment Phases

The equipment phases of PharmaSuite represent a collection of phases that can be used for various recipe or workflow purposes within the framework of PharmaSuite for Production Execution. They provide functions to track the use of individual pieces of equipment throughout the process and to integrate with the equipment automation layer to download, upload, and monitor process-relevant data of the equipment.

The following phases are available:

- Phases for equipment tracking
 - Identify Equipment (page [19](#))
 - Change Equipment Status (page [33](#))
 - Trigger Graph Transition (page [41](#))
 - Separate Equipment (page [51](#))
 - Unbind Equipment (page [61](#))
- Phases for automation integration
 - Get OPC Values (page [69](#))
 - Set OPC Values (page [87](#))
 - Monitor Numeric Value (page [101](#))
 - Get Alarms (page [111](#))
- Phase for Historian integration
 - Show Historical Data Chart (page [121](#))

This manual contains important information about using the equipment phases in master recipes, workflows, or building blocks. Please read this section carefully, because it provides a solid background for all operations you may wish to perform with your system.

Typographical Conventions

This documentation uses typographical conventions to enhance the readability of the information it presents. The following kinds of formatting indicate specific information:

Bold typeface	Designates user interface texts, such as <ul style="list-style-type: none">■ window and dialog titles■ menu functions■ panel, tab, and button names■ box labels■ object properties and their values (e.g. status).
<i>Italic typeface</i>	Designates technical background information, such as <ul style="list-style-type: none">■ path, folder, and file names■ methods■ classes.
CAPITALS	Designate keyboard-related information, such as <ul style="list-style-type: none">■ key names■ keyboard shortcuts.
Monospaced typeface	Designates code examples.

TIP

Instructions in this manual are based on Windows 7. Select the appropriate commands if you are using a different operating system.

Equipment Tracking

In order to compile a complete record of batch production PharmaSuite provides processes not only for tracking material flows but also for tracking equipment entities. All events in the life cycle of an equipment entity, such as its identification and usage in the context of order or workflow processing are recorded in the entity's logbook, which (if enabled) can be viewed in Data Manager at all times.

Equipment tracking heavily relies on the equipment binding (page 3) as basic concept to ensure that operators can use a specific piece of equipment only within one process at a time. Another essential concept is equipment grouping (page 4), which on the one hand allows assembly and disassembly processes for equipment. On the other hand it allows to create temporary groups of otherwise unrelated pieces of equipment for performing common functions on them, such as cleaning procedures.

Binding of Equipment

An equipment entity or equipment entity group can only be used in the context of one unit procedure at any given time. To ensure the exclusive usage, the **Identify equipment** phase does not only identify an equipment entity, but also binds it to the current unit procedure if all relevant checks have passed successfully. As soon as an equipment entity has been identified (i.e. successfully scanned, but not yet bound) or bound, it is no longer available in the context of any other unit procedure.

Equipment entities are either released automatically upon completion of a unit procedure or they can be released explicitly with the **Unbind equipment** phase.

Grouping of Equipment

An equipment entity group consists of a parent entity and one or more child entities. Thus a child entity that has children is the sub-parent to a sub-group, which itself belongs to a main group that has its main parent.

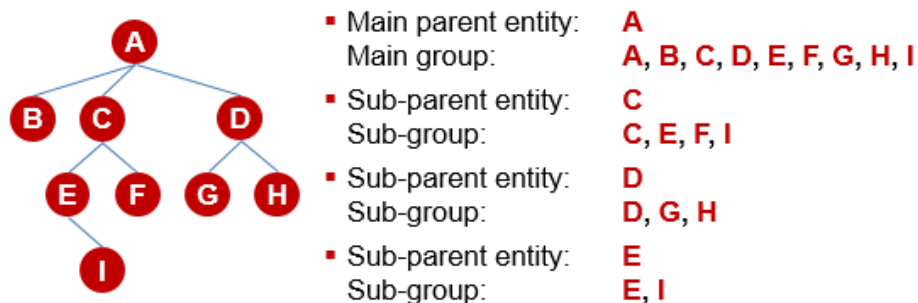


Figure 1: Structure of an equipment entity group

Equipment grouping covers several processing scenarios:

- Setting up a production line or work center.
- Assembling and disassembling equipment.
- Grouping equipment entities that only share a loose connection (e.g. equipment entities that are placed on a tray to be sterilized in an autoclave).

Grouping and ungrouping of equipment entities typically takes place on the shop floor and is considered runtime data. For this reason, an entity's grouping history is listed among the logged events in its logbook, which is maintained in Data Manager. An entity's current grouping situation is also shown in Data Manager, on the **Grouping** tab of the entity. From there, a supervisor can resolve exception situations that have caused inconsistent data or ungroup entities for unscheduled repairs.

In the context of equipment grouping, the phases for equipment tracking can be used for processes such as

- creating an equipment entity group (page 5)
- identifying an equipment entity group for use (page 8)
- re-identifying an already bound equipment entity group (page 8)
- triggering a graph transition on an equipment entity group (page 10)
- separating an equipment entity group (page 11).

Creating an Equipment Entity Group

For creating an equipment entity group, the **Identify equipment** phase is used in two different configuration variants:

- to identify and thus define the parent of the group to be created
 - to identify and thus add a child to the previously identified parent.
- An added child can itself be a parent of a group, which then adds the entire group as sub-group to the main parent.

TIP

Please note that you can create complex group structures within one operation by first creating all required sub-groups and afterwards adding their parents as children to the main parent of the group.

An example of creating a complex group would be the assembly of a tablet press that consists of the tablet press itself as parent, a discharge chute and a feeder group (feeder and feeder seal).

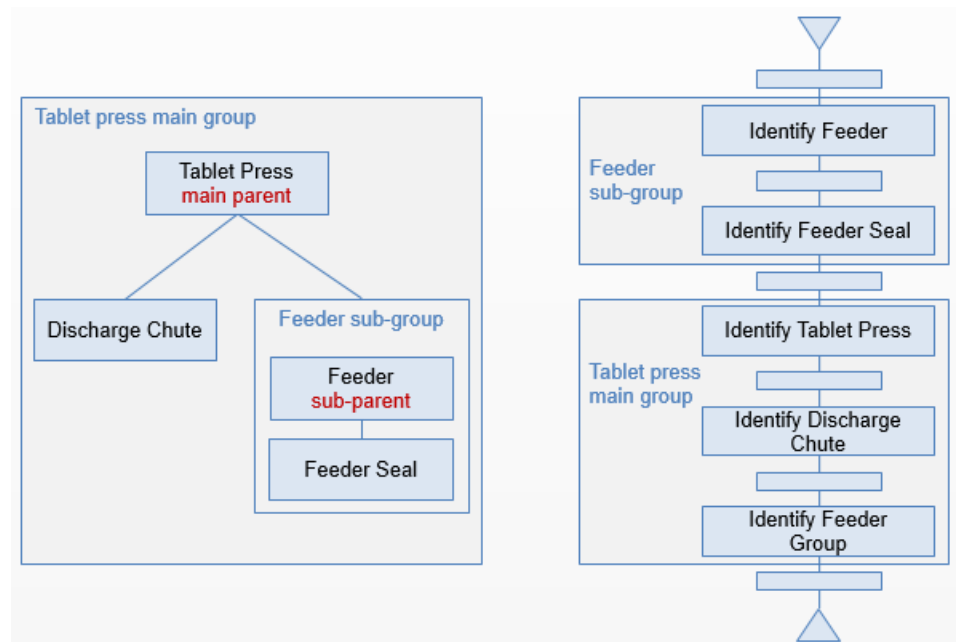


Figure 2: Sample group with assembly operation

The assembly operation consists of a series of **Identify equipment** phases:

Operation phase	Product phase	Purpose and configuration
Identify Feeder	Identify Equipment (RS) (page 19)	<p>Identification of feeder as parent of feeder sub-group:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is not defined. ■ Identify in current binding context parameter (page 25) is not enabled. ■ Identification mode parameter (page 25) is set to Single entity required.
Identify Feeder Seal	Identify Equipment (RS) (page 19)	<p>Identification of feeder seal as child of feeder:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is set to the output of the Identify Feeder phase: <pre>{Identify Feeder}. {Equipment object}</pre> ■ Identify in current binding context parameter (page 25) is not enabled. ■ Identification mode parameter (page 25) is set to Single entity required.
Identify Tablet Press	Identify Equipment (RS) (page 19)	<p>Identification of tablet press as parent of main group:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is not defined. ■ Identify in current binding context parameter (page 25) is not enabled. ■ Identification mode parameter (page 25) is set to Entity or group.

Operation phase	Product phase	Purpose and configuration
Identify Discharge Chute	Identify Equipment (RS) (page 19)	<p>Identification of discharge chute as child of tablet press:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is set to the output of the Identify Tablet Press phase: <code>{Identify Tablet Press}. {Equipment object}</code> ■ Identify in current binding context parameter (page 25) is not enabled. ■ Identification mode parameter (page 25) is set to Single entity required.
Identify Feeder Group	Identify Equipment (RS) (page 19)	<p>Identification of feeder group as sub-group of tablet press:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is set to the output of the Identify Tablet Press phase: <code>{Identify Tablet Press}. {Equipment object}</code> ■ Identify in current binding context parameter (page 25) is enabled. ■ Identification mode parameter (page 25) is set to Entity group required.

Identifying an Equipment Entity Group for Use

To identify an equipment entity group such as an assembled tablet press for use in a production operation, the **Identify equipment** phase is configured as follows:

Operation phase	Product phase	Purpose and configuration
Identify Tablet Press	Identify Equipment (RS) (page 19)	Identification of tablet press assembly for production: <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is not defined. ■ Identify in current binding context parameter (page 25) is not enabled. ■ Identification mode parameter (page 25) is set to Entity or group (if the assembled tablet press is not necessarily tracked as entity group), Entity group required (if the assembled tablet press is tracked as group).

Re-identifying an Already Bound Equipment Entity Group

There are circumstances when an entity that has already been identified in the context of a unit procedure needs to be identified again, for example for grouping or separation purposes or during execution, when it is re-used in another room, but still in the same unit procedure.

To identify an already bound equipment entity that is already bound to the context of the current unit procedure, the **Identify equipment** phase is configured as follows:

Operation phase	Product phase	Purpose and configuration
Identify Measuring Tool	Identify Equipment (RS) (page 19)	<p>Identification of measuring tool that has already been identified and used before in another operation of the process:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is not defined. ■ Identify in current binding context parameter (page 25) is enabled. ■ Identification mode parameter (page 25) is set to Entity or group (if the measuring tool may or may not be an equipment group itself), Single entity required (if the measuring tool must be a single entity), Entity group required (if the measuring tool must have been assembled before and is tracked as group), Child entity (already bound) (if the measuring tool was a member of another group when it was first identified and bound and now needs to be separated from the group).

Triggering a Graph Transition on an Equipment Entity Group

For running a status transition on an entire equipment entity group, the **Trigger graph transition** phase receives the main parent of the group, then checks which of the group members hold a status graph of the configured purpose and performs the selected trigger on those entities. Graphs that do not have the configured purpose are ignored.

TIPS

Please note that at least one entity in the group must hold a graph of the configured purpose. Depending on a graph's configuration, the selected trigger can result in alternative status transitions due to different source statuses of the affected entities or conditions defined with the transitions.

An example of triggering a graph transition on an equipment entity group would be a trigger that is performed after an assembled tablet press has been used. The purpose of the trigger is to define if the assembly is fully available for re-use or if some parts of it need to be cleaned or replaced before the assembled equipment is available again.

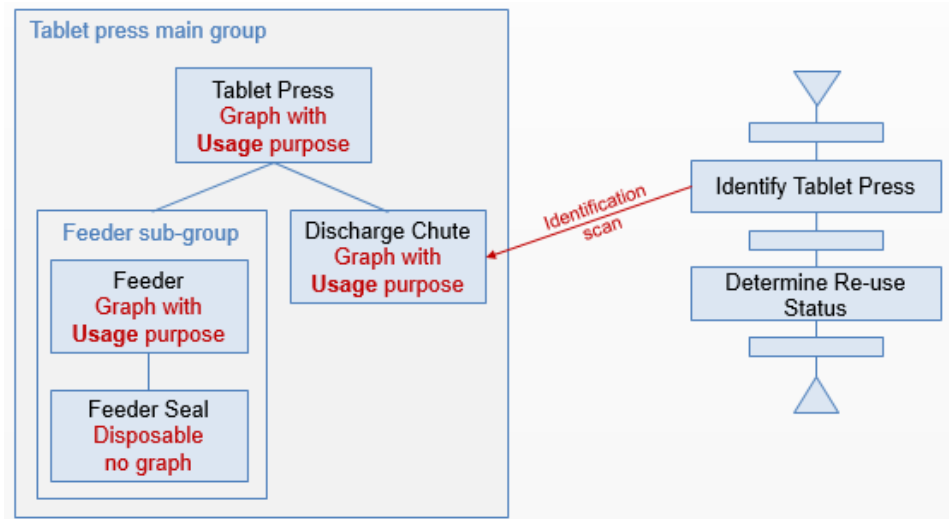


Figure 3: Sample group with re-use status operation

The re-use status operation consists of the following phases:

Operation phase	Product phase	Purpose and configuration
Identify Tablet Press	Identify Equipment (RS) (page 19)	<p>Scan of discharge chute to identify tablet press group:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is not defined. ■ Identify in current binding context parameter (page 25) is not enabled. ■ Identification mode parameter (page 25) is set to Entity group required.
Determine Re-use Status	Trigger Graph Transition (RS) (page 41)	<p>Status change on all entities that hold a status graph with the Usage purpose:</p> <ul style="list-style-type: none"> ■ Identified equipment entity parameter (page 45) is set to the output of the Identify Tablet Press phase: <code>{Identify Tablet Press}. {Equipment object}</code> ■ Allowed triggers parameter (page 25) is set to a suitable trigger of the Usage type.

Separating an Equipment Entity Group

For separating grouped equipment entities the **Separate equipment** phase covers a range of different scenarios. With its process parameters you can configure to fully or partially dissolve groups with or without retaining the binding status of their individual equipment entities. The settings of the following process parameters controls the separation scenarios:

- **Parent entity of group** (page 55)
It must reference the main parent entity of the group.
- **Child entity to be separated** (page 56)
If set, it defines the one specific child entity or sub-group to be separated.
If not set, all child entities and sub-groups are split from the main parent entity.

- **Keep sub-group (page 56)**
If enabled, it retains the grouping of sub-groups that are separated from the main group.
If not enabled, it dissolves the sub-groups into individual entities.
- **Keep binding (page 57)**
If enabled, the separated entities or entity sub-groups retain their binding to the current unit procedure.
If not enabled, the separated entities or entity sub-groups are released from their binding and can be identified for processing in the context of another unit procedure.

Common scenarios and their required parameter settings:

- Separation of an entire group into its individual entities, including the separation of all of its sub-groups, and the release of binding.

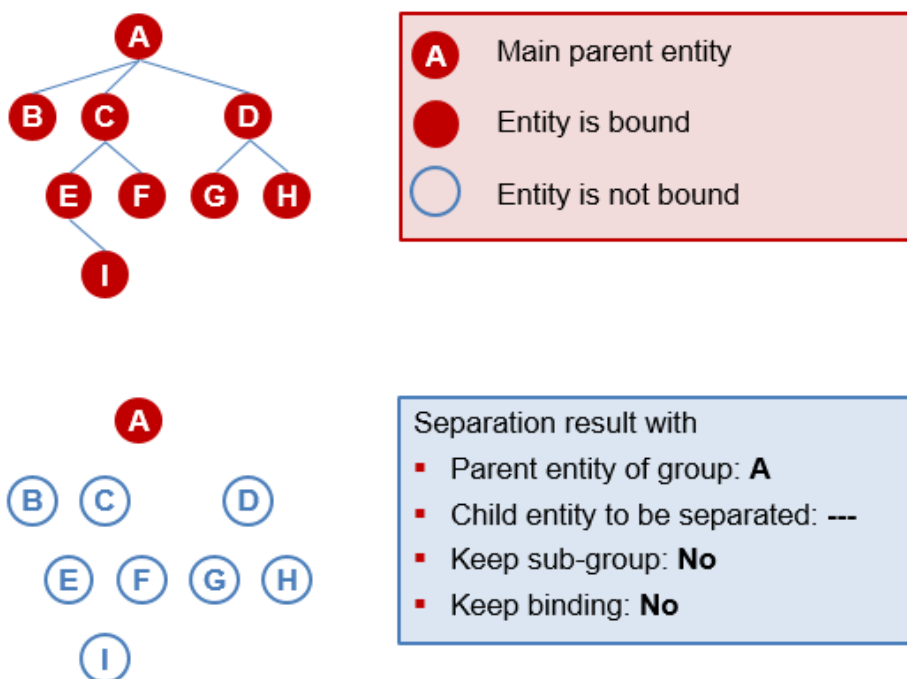
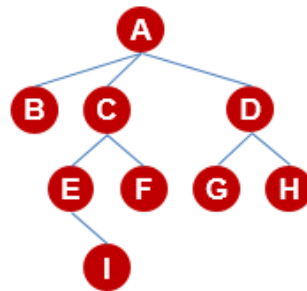
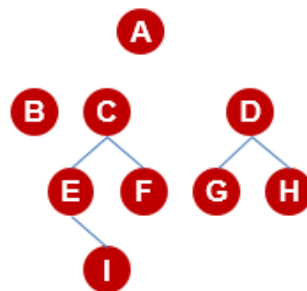


Figure 4: Full separation with release of binding

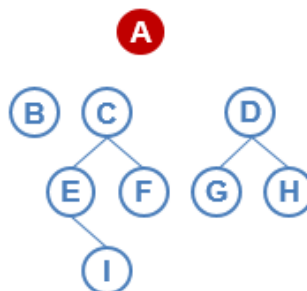
- Separation of all child entities and sub-groups from a main parent, while retaining the grouping of the sub-groups, with or without release of binding.



- A** Main parent entity
- Entity is bound
- Entity is not bound



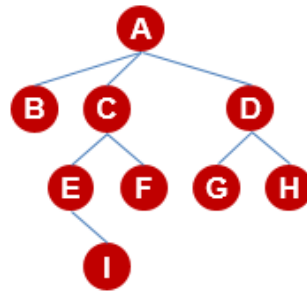
- Separation result with
- Parent entity of group: **A**
 - Child entity to be separated: ---
 - Keep sub-group: **Yes**
 - Keep binding: **Yes**



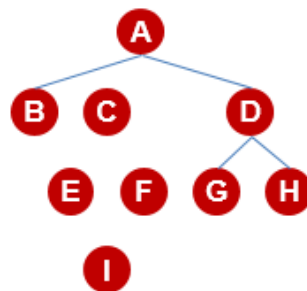
- Separation result with
- Parent entity of group: **A**
 - Child entity to be separated: ---
 - Keep sub-group: **Yes**
 - Keep binding: **No**

Figure 5: Full separation with retained grouping

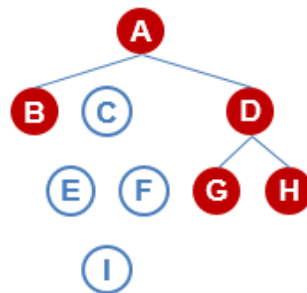
- Separation of one specific sub-group from the main group, including the separation of the sub-group, with or without release of binding.



- A** Main parent entity
- Entity is bound
- Entity is not bound



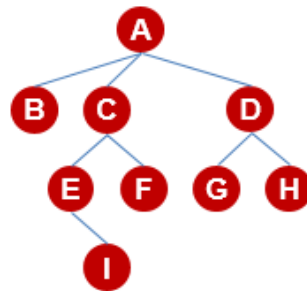
- Separation result with
- Parent entity of group: **A**
 - Child entity to be separated: **C**
 - Keep sub-group: **No**
 - Keep binding: **Yes**



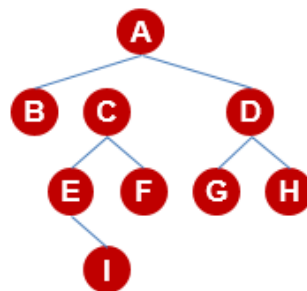
- Separation result with
- Parent entity of group: **A**
 - Child entity to be separated: **C**
 - Keep sub-group: **No**
 - Keep binding: **No**

Figure 6: Partial separation with dissolved grouping

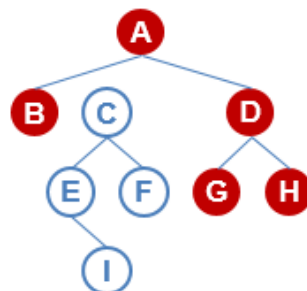
- Separation of one specific sub-group from the main group, while retaining the grouping of the sub-group, with or without release of binding



- A** Main parent entity
- Entity is bound
- Entity is not bound



- Separation result with
- Parent entity of group: **A**
 - Child entity to be separated: **C**
 - Keep sub-group: **Yes**
 - Keep binding: **Yes**



- Separation result with
- Parent entity of group: **A**
 - Child entity to be separated: **C**
 - Keep sub-group: **Yes**
 - Keep binding: **No**

Figure 7: Partial separation with retained grouping

An example of a partial separation with dissolved grouping would be preparing a cleaning procedure on a sub-group (feeder group) of a tablet press.

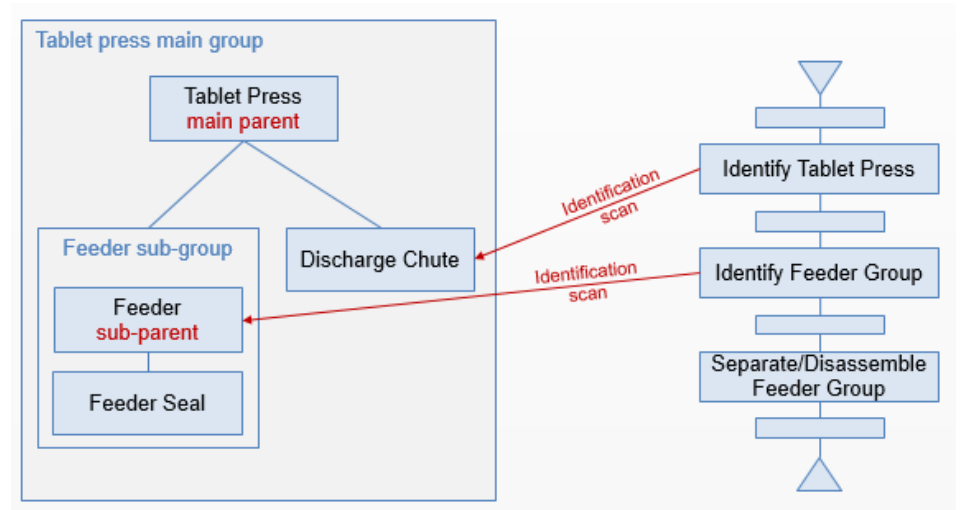


Figure 8: Sample group with operation for partial separation

The separation operation consists of the following phases:

Operation phase	Product phase	Purpose and configuration
Identify Tablet Press	Identify Equipment (RS) (page 19)	<p>Scan of discharge chute to identify tablet press group:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is not defined. ■ Identify in current binding context parameter (page 25) is not enabled. ■ Identification mode parameter (page 25) is set to Entity group required.

Operation phase	Product phase	Purpose and configuration
Identify Feeder Group	Identify Equipment (RS) (page 19)	<p>Scan of feeder to identify the feeder group for separation:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is not defined. ■ Identify in current binding context parameter (page 25) is enabled. ■ Identification mode parameter (page 25) is set to Child entity (already bound).
Separate/Disassemble Feeder Group	Separate Equipment (RS) (page 51)	<p>Separation, disassembly and unbinding of the feeder group:</p> <ul style="list-style-type: none"> ■ Parent entity of group parameter (page 24) is set to the output of the Identify Tablet Press phase: <code>{Identify Tablet Press}. {Equipment object}</code> ■ Child entity to be separated parameter (page 56) is set to the output of the Identify Feeder Group phase: <code>{Identify Feeder Group}. {Equipment object}</code> ■ Keep sub-group parameter (page 56) is not enabled. ■ Keep binding parameter (page 57) is not enabled.

-
-
- Rockwell Software PharmaSuite® BB - User Manual Equipment Phases
-
-

Identify Equipment

The **Identify equipment** phase allows to identify an equipment entity or an equipment entity group and to bind the entity to the context in which it is being used (e.g. order, workflow). Additionally, the phase allows to build an equipment entity group. The identified and bound equipment entity is automatically unbound when the unit procedure is completed as planned or when the order step or workflow is canceled. If the equipment entity needs to be identified again for another phase before the unit procedure has been completed, it can be unbound with the **Unbind equipment** phase (page 61).

It can be used for processing requirements, such as:

- Verifying that an equipment entity or an equipment entity group (parent entity and its child entities) meets requirements
Equipment entities used during processing must meet various requirements. Prior to being used, an equipment entity is checked against the defined requirements (equipment class and additional properties). The ensuing results are documented in the entity's logbook.
- Exclusive usage of an equipment entity or equipment entity group for processing an order
In order to ensure the exclusive usage of a specific equipment entity or equipment entity group, the entity or entity group (including all of its child entities) is bound to a batch and unit. The binding itself is documented in the batch report and the entity's logbook.
- Identification of parent and child equipment entities either for grouping equipment entities or for separating equipment entity groups.

Execution

The **Identify equipment** phase expects an operator to scan a single equipment entity or an equipment entity that is a member of an equipment group for the purpose of using the entity or entity group to process an order or workflow.

Once the operator has scanned an entity, the system performs checks to ensure that

- the single entity or all members of the entity group match the required minimum status.
- the status of the single entity or of all members of the entity group matches the required fit-for-purpose status. For this reason, the system first updates the current status of the single entity or of all members of the entity group to determine if the status has expired, triggers a status transition if that is the case, and then performs the status check on the newly established status.
- the single entity or all members of the entity group match the required equipment class and all additional requirements defined during phase design.
- the required equipment class to which the single entity or the parent entity of the group belong matches the required minimum status.

TIP

Please note that the required minimum status is also checked for classes that are referenced with the **Equipment is Member of Class** function of the flexible rules.

If the identified entity belongs to an entity group, the phase identifies the group as a whole and displays the group parent as identified entity, regardless of which group member the operator has scanned. Additionally, with the **Show children** process parameter (page 26), the phase can be configured to show a list of all children of the displayed parent entity with

- the identifier of each child entity
- the short description of each child entity
- the status values of each child entity.

As long as the phase is active, it provides user-triggered exceptions

- to identify an equipment entity manually, by typing its barcode string
- to revoke the identification of a single equipment entity or an entity group that was erroneously identified
- to skip the identification and complete the phase without identified entity or entity group.

After completion, the phase displays the following data in the Execution Window:

- The identifier of the group parent, if the entity was identified to add it to or remove it from a group.
- The identified equipment entity with
 - the identifier and short description of the equipment class to which the equipment entity belongs
 - the identifiers or descriptions (if available) of the rules assigned to the class as additional requirements
 - the identifier and short description of the equipment entity
 - the status values of the equipment entity.
- The list of children of the identified entity (if configured) with
 - the identifier of each child entity
 - the short description of each child entity
 - the status values of each child entity.

The navigator displays the identifier of the identified equipment entity for a single entity. For an entity group, it displays the identifier of the parent entity and, in a second line, the identifier of the entity that was scanned.


Scan a tableting unit label.			
Required class	Additional requirements	Actual entity	Statuses
TabletPresses-A / Automated Tablet Presses	- Tablet Count / Tablet Count: ea		
		TabletPress_AM / Automated Tablet Press	- Unit Cleaning Graph / Clean
			

Figure 9: Identify equipment during execution


Scan a tableting unit label.			
Required class	Additional requirements	Actual entity	Statuses
TabletPresses-A / Automated Tablet Presses	- Tablet Count / Tablet Count: ea		
		TabletPress_AM / Automated Tablet Press	- Unit Cleaning Graph / Clean
List of child entitles of TabletPress_AM / Automated Tablet Press			
Identifier	Short description	Statuses	
TabletDischargeChute	Discharge chute for Automated Tablet Press	- Unit Cleaning Graph / Clean	
TabletPressFeeder	Feeder for Automated Tablet Press	- Unit Cleaning Graph / Clean	
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press	- Unit Cleaning Graph / Clean	
			

Figure 10: Identify equipment group during execution

Scan a feeder label.

Parent entity for grouping: TabletPress_AM / Automated Tablet Press

Required class	Additional requirements	Actual entity	Statuses
TabletPresses-A / Automated Tablet Presses	- Rule-01 / Feeder group clean	TabletPressFeeder / Feeder for Automated Tablet Press	- Unit Cleaning Graph / Clean

List of child entities of TabletPressFeeder / Feeder for Automated Tablet Press

Identifier	Short description	Statuses
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press	- Unit Cleaning Graph / Clean


Confirm 

Figure 11: Identify equipment group for grouping during execution

Identify by typing the entity barcode.

Confirm 

Undo the identification of the entity. TabletPress_AM

Confirm 

Skip the identification.

Confirm 

Figure 12: User-triggered exceptions of Identify equipment

Scan a tableting unit label.

Required class	Additional requirements	Actual entity	Statuses
TabletPresses-A / Automated Tablet Presses	- Tablet Count / Tablet Count: ea	TabletPress_AM / Automated Tablet Press	- Unit Cleaning Graph / Clean


Confirm 

Figure 13: Identify equipment after phase completion

Scan a tableting unit label.

Required class	Additional requirements	Actual entity	Statuses
TabletPresses-A / Automated Tablet Presses	- Tablet Count / Tablet Count: ea	TabletPress_AM / Automated Tablet Press	- Unit Cleaning Graph / Clean

List of child entities of TabletPress_AM / Automated Tablet Press

Identifier	Short description	Statuses
TabletDischargeChute	Discharge chute for Automated Tablet Press	- Unit Cleaning Graph / Clean
TabletPressFeeder	Feeder for Automated Tablet Press	- Unit Cleaning Graph / Clean
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press	- Unit Cleaning Graph / Clean


Confirm 

Figure 14: Identify equipment group after phase completion

Scan a tableting unit label.

Required class	Additional requirements	Actual entity	Statuses
TabletPresses-A / Automated Tablet Presses	- Tablet Count / Tablet Count: ea		
		TabletPress_AM / Automated Tablet Press	- Unit Cleaning Graph / Clean

List of child entities of TabletPress_AM / Automated Tablet Press

Identifier	Short description	Statuses
TabletDischargeChute	Discharge chute for Automated Tablet Press	- Unit Cleaning Graph / Clean
TabletPressFeeder	Feeder for Automated Tablet Press	- Unit Cleaning Graph / Clean
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press	- Unit Cleaning Graph / Clean


Confirm 

Figure 15: Identify equipment group for grouping after phase completion

Identification Identify Tableting Unit	TabletPress_AM
---	----------------

Figure 16: Identify equipment in the Navigator

Identification Identify Tableting Unit	TabletPress_AM TabletDischargeChute
---	--

Figure 17: Identify equipment group in the Navigator

Phase Design

The characteristics of the **Identify equipment** phase are defined via process parameters and their attributes.

Its user interface is designed in two columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions. If the entity is identified to add it to or remove it from a group, the second row displays the identifier of the group's parent entity. In the next rows the phase displays a tabular view of relevant data of the equipment entity, spanning all columns. Once an equipment entity has been identified, its data appears as new row of the table. If configured to show, the following rows display the list of children of the identified entity, spanning all columns. The right column in the bottom row provides the **Confirm** button.

When the phase is completed it shows the same two-column, multi-row layout with the data of the identified equipment entity listed in the table as well as the table for the list of children, if configured.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Text	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).

Mode

Defines if the phase expects operator interaction during execution.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. Manual completion (default): Operator confirms the phase. Automatic completion : Phase automatically sets the property values and is completed.

Parent entity of group

TIP

Please note that indicating a parent entity is only necessary for processes concerned with grouping or ungrouping of equipment entities. For all other processes, the **Parent entity of group** process parameter is not relevant and must remain blank.

Indicates the parent of the entity group to which you wish to add the identified entity or entity group. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the equipment object output variable of a preceding phase that defines the parent entity of the group. The identified equipment entity is added to the parent entity as a child entity.

Identify in current binding context

Defines if the equipment entity must have been identified before in the process. This is the case when a grouped equipment entity has been identified along with its group and need to be specifically identified again as single entity in order to remove it from the group.

Attribute	Type	Comment
Enabled	Boolean	Controls if the identification is only possible if the entity or entity group is already bound to the current unit procedure context.

Identification mode

Defines which grouping situation is required for the identified entity. If you select the **Child entity (already bound)** option, make sure that you have enabled the **Identify in current binding context** option (page 25).

Attribute	Type	Comment
Enabled	Choice list	Defines the identification mode. Available settings: Entity or group , Single entity required , Entity group required , Child entity (already bound) . Default setting: Entity or group .

Show children

Defines whether it is relevant to display a table that lists all child entities of the identified entity.

TIP

Please note that the table is not displayed if the identified entity does not have any children, even if you have enabled the **Show children** option.

Attribute	Type	Comment
Enabled	Boolean	Controls if the list of child entities is displayed during execution.

Property value check

Represents a system-triggered exception that is displayed in the Exception Window to indicate why and in which properties a scanned equipment entity does not meet the pre-defined equipment requirements.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Equipment status check

Represents a system-triggered exception that is displayed in the Exception Window. It enables the use of not yet approved equipment for qualification purposes.

Attribute	Type	Comment
Minimum class status	Choice list	Defines the minimum class status required for equipment identification. Available settings: Verification , Approved . Default setting: Approved .
Minimum entity status	Choice list	Defines the minimum entity status required for equipment identification. Available settings: Verification , Approved . Default setting: Approved .
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Enter identifier manually

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to enter the barcode string of the required equipment entity manually.

It covers incidents when it is not possible to scan the respective label.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Undo identification

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to revoke the identification of a previously identified equipment.

It covers incidents when an equipment entity is accidentally contaminated after identification, is thus rendered unusable and needs to be replaced with another suitable equipment entity.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Skip identification

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to confirm the phase without identified equipment entity. It covers incidents when there is no suitable equipment available to be identified, but processing can continue regardless, for example when identification is run in a loop to group entities for a cleaning process.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Identify equipment** phase provides the following output variables:

Equipment ID

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Coater-A**" or "**In process**".
- Usage: The output variable provides the identifier of the identified equipment entity for displaying it as text.

Equipment short description

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Coater-A**" or "**In process**".
- Usage: The output variable provides the short description of the identified equipment entity for displaying it as text.

Equipment object

- Data type: IMESS88Equipment, used for transporting an entire equipment data object with all its information and references to provide as process input.
- Usage: The output variable provides the complete object of the identified equipment entity. This is the output to use in subsequent phases for accessing data of the equipment object, such as changing its status or writing a property.

Result

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Coater-A**" or "**In process**".
- Usage: The output variable states if an equipment entity has been identified:
 - The value is `SKIPPED` if the phase has been completed without identifying an equipment entity.
 - The value is `IDENTIFIED` if an equipment entity has been identified and thus bound.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString  
  (convertTo  
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},  
    "min")  
  )  
  + " min"
```

As result of the expression, the system displays **"92 min"**.

Change Equipment Status

The **Change equipment status** phase allows an operator to set an FSM-related status of a previously identified equipment entity.

TIP

Please note that the phase neither supports status changes on Data Manager-maintained status graphs nor status changes on equipment entity groups.

It can be used for processing requirements, such as:

- Need to set the equipment entity's status
Depending on the current process step, the status must be changed from **Clean** to **In use**.

Execution

The **Change equipment status** phase lists all status change actions that are available from the present status of the equipment entity to which the phase refers. An operator can select a suitable action and thus change the entity's status.

As long as the phase is active, it provides a user-triggered exception to force a specific equipment status that could not be reached by an available status change action.

After completion, the phase displays the affected equipment entity with the following data in the Execution Window:

- Identifier and short description of the equipment entity
- Identifier of the FlexibleStateModel property that governs the statuses of the equipment entity
- Status change action that was performed
- New status of the equipment entity.

The navigator displays the identifier of the affected equipment entity.

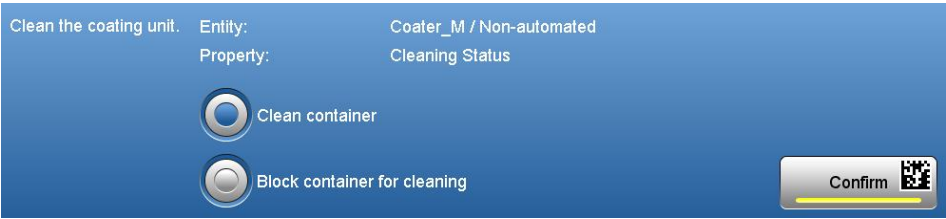


Figure 18: Change equipment status during execution

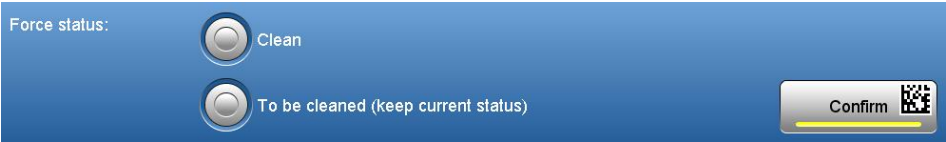


Figure 19: User-triggered exception of Change equipment status

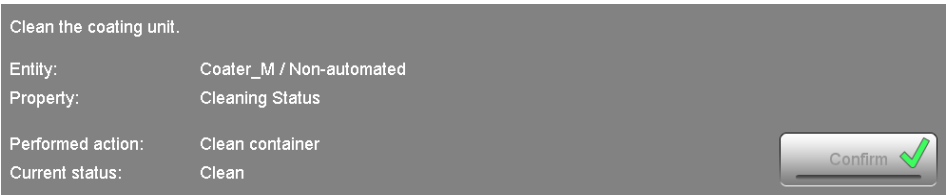


Figure 20: Change equipment status after phase completion



Figure 21: Change equipment status in the Navigator

Phase Design

The characteristics of the **Change equipment status** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions. The second column displays the identifier and short description of the entity as well as the identifier of the FSM property that controls the entity's statuses.

In the next rows the phase displays a list of options, one for each of the defined potential change actions. The right column in the bottom row provides the **Confirm** button.

TIP

Please note that the number of available change actions shown during execution may differ from what you see during phase definition in the preview. This happens since the system only displays those change actions as options that the FSM allows for the actual status of the identified entity.

When the phase is completed it shows a three-column, three-row layout with the textual instruction taking up the first row, followed by a listed display of the data of

- the identified equipment entity,
- the FSM property,
- the performed status change action, and
- the new status the equipment entity has reached.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Not used.
Column 3	HTML text	Not used.

Identified equipment entity

Indicates the equipment entity to be processed with the phase. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the output of a preceding phase that provides an identified equipment entity.

Allowed change actions

Defines the change actions available for selection during execution. You need to select all possible change actions for all statuses the identified equipment entity may have. During execution, the system takes the entity's actual status into account and only displays those options that comply with the rules of the FlexibleStateModel.

To reach another one of the defined allowed statuses that would otherwise not be available or to retain the entity's current status, the operator has to file a **Force status** exception (page 37).

Attribute	Type	Comment
Property and allowed change actions	Text (structured)	The property that needs to match the respective status property of the identified equipment entity on the shop floor and the allowed actions from the list of all supported actions of the selected property.

For selecting the allowed change actions from the list of change actions defined for the FSM, the system provides a Change Action Selection editor.

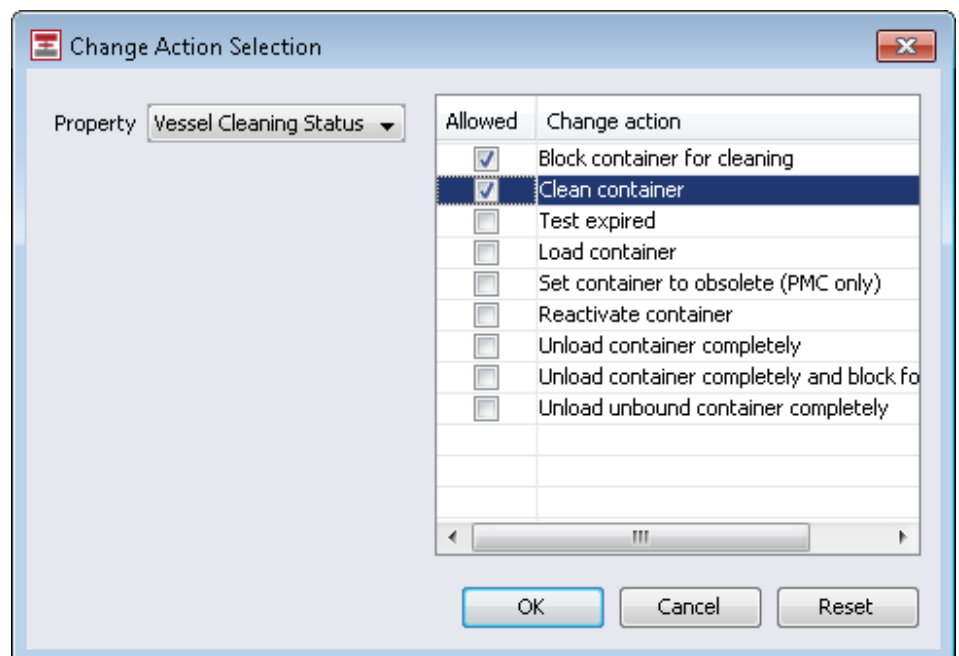


Figure 22: Change Action Selection editor

Equipment data mismatch

Represents a system-triggered exception that is displayed in the Exception Window to indicate why and in which properties the identified equipment entity does not meet the pre-defined equipment requirements.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Force status

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to set the identified equipment entity to a specific status even though the FSM does not allow this change action. It lists all statuses reached by the selection made in the Allowed change actions (page 36) parameter, including an option that allows the operator to retain the status of the entity (**Keep current status**). It covers incidents when an equipment entity's status would normally render it unsuitable for a planned status change, but the issue has been resolved by other means that are not transparent to the system.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Change Equipment Status** phase provides the following output variables:

Status change result

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**GOOD**" or "**PLANNED**".
- Usage: The output variable states the conditions under which the status change has been performed.
 - The value is **PLANNED** if the status change has been performed as defined in the phase.
 - The value is **FORCED** if the status change has been forced by the operator.
 - The value is **SKIPPED** if no status change has been performed at all (**Keep current status** option).

New status

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Coater-A**" or "**In process**".
- Usage: The output variable provides the new status of the equipment entity.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString  
  (convertTo  
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},  
    "min")  
  )  
  + " min"
```

As result of the expression, the system displays **"92 min"**.

Trigger Graph Transition

The **Trigger graph transition** phase allows an operator to change an equipment graph-related status of a previously identified equipment entity or equipment entity group (parent entity and its child entities) by executing a graph trigger.

It can be used for processing requirements, such as:

- Need to trigger a graph transition on the equipment entity
Depending on entity properties like counter, dates, and the current status, a trigger causes a status transition from **In use** to **Reusable** or **Uncleaned**.

Execution

The **Trigger graph transition** phase lists selected transition triggers of a specific purpose, which can be performed to change the status of the equipment entity or entity group to which the phase refers. An operator can select a suitable trigger and thus change the status of the entity or of all entities of the group, respectively.

TIP

Please note that an entity group is represented by the identifier of its parent entity.

If the phase refers to an entity group, it can be configured with its **Show children** process parameter (page 45) to show a list of all children of the displayed parent entity with

- the identifier of each child entity
- the short description of each child entity
- the status values of each child entity.

After completion, the phase displays the affected equipment entity with the following data in the Execution Window:

- Identifier and short description of the equipment entity or the parent of the equipment group, respectively
- Equipment graph that governs the statuses of the equipment entity or the parent of the equipment group, respectively
- Trigger that was performed
- Old status of the equipment entity

- New status of the equipment entity
- The list of the affected children of the identified entity (if configured) with
 - the identifier of each child entity
 - the short description of each child entity
 - the graphs and status values of each child entity.

The navigator displays the identifier of the affected equipment entity for a single entity. For an entity group, it displays the identifier of the parent entity and, in a second line, the number of entities whose status was changed.

Change the cleaning status of the room according to the performed cleaning procedure.

Entity: R-001 (D) / Weighing booth
 Graph: Room Cleaning (RS) [General Cleaning Required]

☒ Cleaning (general)
☐ Cleaning (major)
☐ Cleaning (minor)

Confirm

Figure 23: Trigger graph transition during execution

Set the cleaning status for the sterilized components.

Entity: SterilizationTray / Feeder for Automated Tablet Press
 Graph: Unit Cleaning Graph [To be cleaned]

List of child entities of SterilizationTray / Feeder for Automated Tablet Press

Identifier	Short description	Statuses
TabletDischargeChute-AS	Discharge chute for Automated Tablet Press	- Unit Cleaning Graph / To be cleaned
TabletPressFeeder-AS	Feeder for Automated Tablet Press	- Unit Cleaning Graph / To be cleaned
TabletPressFeederSeal-AS	Feeder Seal for Automated Tablet Press	- Unit Cleaning Graph / To be cleaned

☒ Cleaned

Confirm

Figure 24: Trigger graph transition of group during execution

Change the cleaning status of the room according to the performed cleaning procedure.

Entity: R-001 (D) / Weighing booth
 Graph: Room Cleaning (D)

Performed trigger: Cleaning (general)
 Old status: General cleaning required
 New status: Clean (general)

Confirm

Figure 25: Trigger graph transition after phase completion

Set the cleaning status for the sterilized components.

Entity: SterilizationTray / Feeder for Automated Tablet Press
 Graph: Unit Cleaning Graph

Performed trigger: Cleaned
 Old status: To be cleaned
 New status: Clean

List of child entities of SterilizationTray / Feeder for Automated Tablet Press

Identifier	Short description	Statuses
TabletDischargeChute-AS	Discharge chute for Automated Tablet Press	- Unit Cleaning Graph / Clean
TabletPressFeeder-AS	Feeder for Automated Tablet Press	- Unit Cleaning Graph / Clean
TabletPressFeederSeal-AS	Feeder Seal for Automated Tablet Press	- Unit Cleaning Graph / Clean

Figure 26: Trigger graph transition of group after phase completion

Room Cleaning (D)
Change Room Status

R-001 (D)

Figure 27: Trigger graph transition in the Navigator

Sterilization Process
Run Sterilization

SterilizationTray
(3 child entities)

Figure 28: Trigger graph transition of group in the Navigator

Phase Design

The characteristics of the **Trigger Graph Transition** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions. In the next two rows, the first column displays the labels for the entity and graph data shown in the center column. If configured to show, the following rows display the list of children of the identified entity, spanning all columns.

In the next rows the phase displays a list of options, one for each of the defined triggers. The right column in the bottom row provides the **Confirm** button.

TIP

Please note that the phase only displays trigger keys in the preview. Only when the phase has become active, does it show the display text defined for the trigger of the graph that governs the identified entity.

If you have defined a trigger option that is not part of the entity's graph, the option continues to show the trigger key, followed by (N/A).

If the trigger is to be performed on an entity group and you have defined a trigger option that is not available in the graphs of all members of the group, the option shows the trigger key or display text, followed by (incomplete).

When the phase is completed it shows a three-column, multi-row layout with the textual instruction taking up the first row, followed by the listed display of the phase data of the new status the equipment entity has reached.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Text	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).

Mode

Defines if the phase expects operator interaction during execution.

TIP

When there is only one trigger available, the phase performs the trigger and completes automatically, regardless of the setting of the **Mode** parameter.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. Manual completion (default): Operator confirms the phase. Automatic completion : Phase automatically sets the property values and is completed.

Identified equipment entity

Indicates the equipment entity to be processed with the phase. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the output of a preceding phase that provides an identified equipment entity.

Show children

Defines whether it is relevant to display a table that lists all child entities of the identified entity that hold the configured graph.

TIP

Please note that the table is not displayed if the identified entity does not have any children, even if you have enabled the **Show children** option.

Attribute	Type	Comment
Enabled	Boolean	Controls if the list of child entities is displayed during execution.

Allowed triggers

Defines the triggers available for selection during execution. On the basis of the available graph purposes, you need to select the triggers for all transitions to another status that may need to be performed on the identified equipment.

TIP

Please note the equipment an operator eventually identifies during execution needs to have a graph of the same purpose assigned to it as the purpose you use for selecting the triggers.

Attribute	Type	Comment
Purpose and allowed triggers	Text (structured)	The system determines which of the identified entity's assigned graphs matches the purpose of the selected triggers and shows the display texts of the triggers as options.

For selecting the allowed triggers from the list of triggers available for the selected purpose, the system provides a Trigger Selection editor, which shows a warning marker for triggers that are not contained in all graphs of the selected purpose.

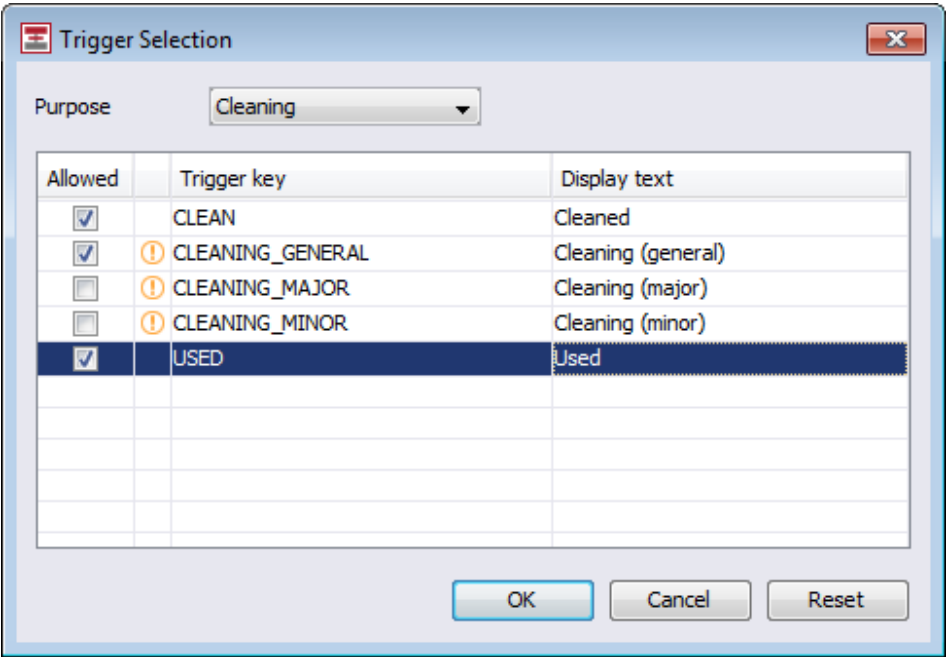


Figure 29: Trigger Selection editor

Equipment data mismatch

Represents a system-triggered exception that is displayed in the Exception Window to indicate why and in which properties or graphs the identified equipment entity does not meet the pre-defined equipment requirements.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Status transition failed

Represents a system-triggered exception that is displayed in the Exception Window to indicate due to which graph attributes the status transition failed.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Trigger graph transition** phase provides the following output variables:

Trigger execution successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the result of the equipment graph trigger execution was successful.
 - The value is `true` if the graph transition with the selected trigger has been executed successfully.
 - The value is `false` if the phase is completed with an exception.

Graph

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Coater-A**" or "**In process**".
- Usage: The output variable provides the identifier of the used equipment graph. It applies to a single entity or to a main parent entity; it does not represent the graph of a child entity.

Selected trigger

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Coater-A**" or "**In process**".
- Usage: The output variable provides the trigger (key) of the used equipment graph.

Old status

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Coater-A**" or "**In process**".
- Usage: The output variable provides the old status (key) of the equipment entity. It applies to a single entity or to a main parent entity; it does not represent the status of a child entity.

New status

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Coater-A**" or "**In process**".
- Usage: The output variable provides the new status (key) of the equipment entity. It applies to a single entity or to a main parent entity. It does not represent the status of a child entity.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as **"Read Instruction"**.
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString  
  (convertTo  
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},  
    "min")  
  )  
  + " min"
```

As result of the expression, the system displays **"92 min"**.

Separate Equipment

The **Separate equipment** phase allows an operator to separate an equipment entity group. Either the entire group is separated into single entities or a single entity or a sub-group is separated from a main group.

Based on the phase configuration, a separated equipment entity can keep its binding status and a separated sub-group can keep its grouping by retaining its relations to its own child entities.

It can be used for processing requirements, such as:

- Need to separate a coater before cleaning
A coater has to be separated into single equipment entities to run specific cleaning processes for different parts.

Execution

The **Separate equipment** phase displays all data configured for the entity separation to be performed:

- The identifier and short description of the main parent entity of the group to be separated
- The identifier and short description of the single entity or entity group to be separated from the parent, if configured
- The setting of the option that defines if a separated sub-group is fully dissolved
- The setting of the option that defines if the separated entity or entities remain bound to the context of their current unit procedure
- The list of all entities to be separated from the main parent with
 - the identifier of each child entity
 - the short description of each child entity.

TIP

Please note that an entity group is represented by the identifier of its parent entity.

As long as the phase is active, it provides a user-triggered exception to skip the separation and complete the phase without having separated the equipment group.

After completion, the phase displays the affected equipment entity with the following data in the Execution Window:

- The identifier and short description of the main parent entity of the separated group
- The identifier and short description of the single entity or entity group that were separated from the parent, unless the entire group was split into individual entities
- The definition if a separated sub-group was fully ungrouped
- The definition if the separated entity or entities remained bound to the context of their current unit procedure
- The list of all entities separated from the main parent with
 - the identifier of each child entity
 - the short description of each child entity.

The navigator displays the identifier of the main parent entity of the separated group. If a single entity is separated from the main group, its identifier is shown in the second row of the navigator. If an entity group was separated from the main group, the second row contains the identifier of the sub-parent. If the entire group was split into individual entities, the navigator does not display a second row.

Remove the feeder seal from the tablet press.

Parent entity of group: TabletPress_AM / Automated Tablet Press

Entity to be separated: TabletPressFeederSeal / Feeder Seal for Automated Tablet Press

Keep sub-group: No

Keep binding: No

Entities to be separated

Identifier	Short description
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press


Confirm 

Figure 30: Separate equipment (single entity) during execution

Remove the feeder group from the tablet press.

Parent entity of group: TabletPress_AM / Automated Tablet Press

Entity to be separated: TabletPressFeeder / Feeder for Automated Tablet Press

Keep sub-group: Yes

Keep binding: No

Entities to be separated

Identifier	Short description
TabletPressFeeder	Feeder for Automated Tablet Press
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press

Confirm 

Figure 31: Separate equipment (entity group) during execution

Disassemble the tablet press.

Parent entity of group: TabletPress_AM / Automated Tablet Press

Keep sub-group: No

Keep binding: No

Entities to be separated

Identifier	Short description
TabletDischargeChute	Discharge chute for Automated Tablet Press
TabletPressFeeder	Feeder for Automated Tablet Press
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press

Confirm 

Figure 32: Separate equipment (all entities) during execution

Skip the separation.

Confirm 

Figure 33: User-triggered exception of Separate equipment

Remove the feeder seal from the tablet press.

Parent entity of group: TabletPress_AM / Automated Tablet Press

Entity to be separated: TabletPressFeederSeal / Feeder Seal for Automated Tablet Press

Keep sub-group: No

Keep binding: No

Separated entities

Identifier	Short description
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press


Confirm 

Figure 34: Separate equipment (single entity) after phase completion

Remove the feeder group from the tablet press.

Parent entity of group: TabletPress_AM / Automated Tablet Press

Entity to be separated: TabletPressFeeder / Feeder for Automated Tablet Press

Keep sub-group: Yes

Keep binding: No

Separated entities

Identifier	Short description
TabletPressFeeder	Feeder for Automated Tablet Press
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press


Confirm 

Figure 35: Separate equipment (entity group) after phase completion

Disassemble the tablet press.

Parent entity of group: TabletPress_AM / Automated Tablet Press

Keep sub-group: No

Keep binding: No

Separated entities

Identifier	Short description
TabletDischargeChute	Discharge chute for Automated Tablet Press
TabletPressFeeder	Feeder for Automated Tablet Press
TabletPressFeederSeal	Feeder Seal for Automated Tablet Press


Confirm 

Figure 36: Separate equipment (all entities) after phase completion

Disassembly Remove Feeder Seal

TabletPress_AM
TabletPressFeederSeal

Figure 37: Separate equipment (single entity) in the Navigator

Disassembly Remove Feeder Group

TabletPress_AM
TabletPressFeeder

Figure 38: Separate equipment (entity group) in the Navigator

Disassembly Disassemble Tablet Press

TabletPress_AM

Figure 39: Separate equipment (all entities) in the Navigator

Phase Design

The characteristics of the **Separate Equipment** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions. In the next rows, the first column displays the labels for the configuration data shown in the center column. The following rows display the list if entities to be separated, spanning all column. The right column in the bottom row provides the **Confirm** button.

When the phase is completed it shows the same three-column, multi-row layout with the textual instruction taking up the first row, followed by the listed display of the phase data and the tabular view of the separated entities.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Text	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).

Mode

Defines if the phase expects operator interaction during execution.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. Manual completion (default): Operator confirms the phase. Automatic completion : Phase automatically separates the equipment entities and is completed.

Parent entity of group

Defines the main parent of the group from which either a single entity or an entity sub-group needs to be split off or that needs to be fully separated into its individual entities or sub-groups.

You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the equipment object output variable of a preceding phase that provides the main parent entity of a group.

Child entity to be separated

Defines the single entity or the parent of the sub-group to be split off from the main group. If the entire main group needs to be fully separated into its individual entities or sub-groups, the process parameter is not relevant and must not be configured. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the equipment object output variable of a preceding phase that provides the child entity to be separated from a group.

Show children

Defines whether it is relevant to display a table that lists all child entities to be separated from the identified entity.

TIP

Please note that the table is not displayed if the identified entity does not have any children, even if you have enabled the **Show children** option.

Attribute	Type	Comment
Enabled	Boolean	Controls if the list of child entities is displayed during execution.

Keep sub-group

Defines if the sub-group whose parent is indicated in the **Child entity to be separated** process parameter (page 56) is retained as group after it has been separated from the main group.

If only a single entity is to be separated or the main group is to be fully separated into individual entities, the option must not be selected.

Attribute	Type	Comment
Enabled	Boolean	Controls if any sub-groups (if applicable) of a separated entity are kept. If not, all sub-groups of the separated entity are completely separated as well.

Keep binding

Defines if the separated entities or entity groups remain bound to the context of their current unit procedure. This prevents the equipment from being identified for use for another order or workflow. The binding is released automatically when the unit procedure is completed.

Attribute	Type	Comment
Enabled	Boolean	Controls if separated entities or sub-groups keep their binding status. If not, they are unbound along with the separation.

Skip separation

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to confirm the phase without having performed the separation. It covers incidents when the label of an entity to be separated cannot be scanned for external reasons, and the phase thus misses required information. If the separation is performed regardless, the grouping data of the affected entities needs to be adjusted in Data Manager to match the real situation on the shop floor.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Separate equipment** phase provides the following output variables:

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString
(
  convertTo
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
    "min")
)
+ " min"
```

As result of the expression, the system displays "92 min".

Unbind Equipment

The **Unbind equipment** phase allows an operator to release a previously bound equipment entity or all equipment entities of a group before a unit procedure is finished.

It can be used for processing requirements, such as:

- Need to explicitly unbind an equipment entity.
A measuring tool needs to be used in multiple orders that run in parallel. It must be released from one order prior to the automatic release along with order step finish. Once released, it can again be identified and used in a different order.
- Need to explicitly unbind an equipment group
Cleaning of a used machine with all of its components has to be started before the current order step is completed. The machine can be unbound as soon as it is no longer used in the context of the current order step.

Execution

The **Unbind equipment** phase displays the identifier and short description of the equipment entity or entity group to which the phase refers. The entity or group is released from its order or workflow context when the operator confirms the phase.

TIP

Please note that an entity group is represented by the identifier of its parent entity.

If the phase refers to an entity group, it can be configured with its **Show children** process parameter (page 45) to show a list of all children of the displayed parent entity with

- the identifier of each child entity
- the short description of each child entity.

After completion, the phase displays the following data in the Execution Window:

- The identifier and short description of the unbound equipment entity
- The list of children of the identified entity (if configured) with
 - the identifier of each child entity
 - the short description of each child entity.

The navigator displays the identifier of the unbound equipment entity for a single entity. For an entity group, it displays the identifier of the parent entity and, in a second line, the number of entities that were unbound.



Figure 40: Unbind equipment during execution

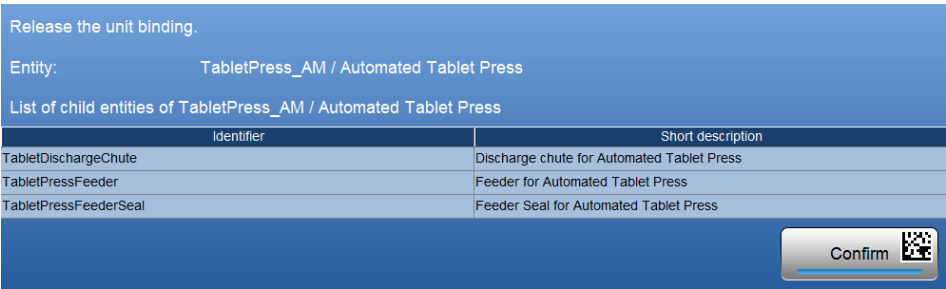


Figure 41: Unbind equipment group during execution

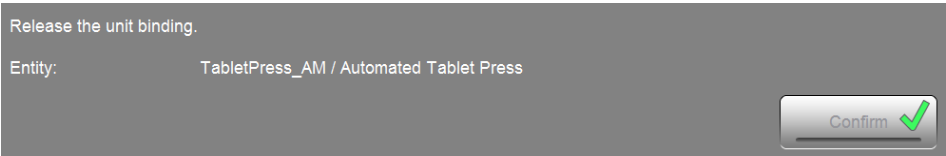


Figure 42: Unbind equipment after phase completion

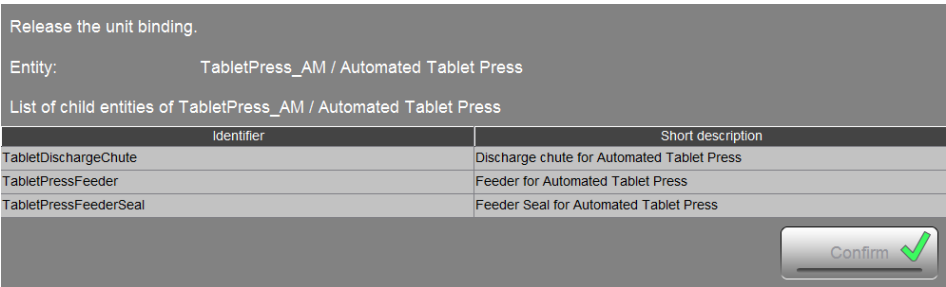


Figure 43: Unbind equipment group after phase completion



Figure 44: Unbind equipment in the Navigator



Figure 45: Unbind equipment group in the Navigator

Phase Design

The characteristics of the **Unbind equipment** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns. The merged columns of the first row provide space for textual instructions. The second column displays the identifier and short description of the entity to which the phase refers. If configured to show, the following rows display the list of children of the identified entity, spanning all columns.

The rightmost column provides the **Confirm** button.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Text	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).

Identified equipment entity

Indicates the equipment entity to be processed with the phase. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the output of a preceding phase that provides an identified equipment entity.

Mode

Defines which amount of operator interaction the phase expects during execution. Unless the system detects an exception, the phase does not require operator interaction when it is set to **Automatic completion**.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. Manual completion (default): Operator confirms the phase. Automatic completion : Phase automatically unbinds the equipment entities and is completed.

Show children

Defines whether it is relevant to display a table that lists all child entities of the identified entity.

TIP

Please note that the table is not displayed if the identified entity does not have any children, even if you have enabled the **Show children** option.

Attribute	Type	Comment
Enabled	Boolean	Controls if the list of child entities is displayed during execution.

Equipment data mismatch

Represents a system-triggered exception that is displayed in the Exception Window to indicate why and in which properties the identified equipment entity does not meet the pre-defined equipment requirements.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Equipment unbind failed

Represents a system-triggered exception that is displayed in the Exception Window. It provides a way to proceed if there is a technical issue that prevents the unbinding of the equipment entity.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Unbind Equipment** phase provides the following output variables:

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString
(
  convertTo
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
    "min")
)
+ " min"
```

As result of the expression, the system displays "92 min".

Get OPC Values

The **Get OPC values** phase allows to read up to 50 tag values of one equipment entity from the automation layer.

It can be used for processing requirements, such as:

- Verify parameters of a mixer
With one button tap, an operator can retrieve the values of all relevant set points of a mixer from the automation layer to check them against defined limits. Any violation can be tracked as an exception. Finally, the mixer speed is passed on to a subsequent phase for calculation or decision purposes.
- Mixer speed should range between 400 rpm and 1000 rpm.
- HeatingControl should be set to Yes.
- HeatingTargetTemp should be 55 °C.
- HeatingProfile should be 7.
- HMI_InstructionText1 should be "Control visual foam situation".

The phase supports the following data types: Boolean, BigDecimal, and String.

Execution

The **Get OPC values** phase displays the identifier and short description of the equipment entity to which the phase refers. When the operator taps the **Get** button, the phase accesses the automation layer, reads the tag values provided for the specified property, and records a timestamp for each value it has read. For Numeric values, it can be configured to perform limit checks on the retrieved values against two pre-defined sets of limits for low and high (**L/LL** and **H/HH**). For Boolean and String values, the system checks the retrieved values against an expected value. Which of the limits or expected values are actually available and enabled for checking is configured with the phase's process parameters.

During execution, if a value is affected by a technical issue, such as a communication failure, its cell background assumes a different color (red) and displays a marker symbol (X).

Different phase modes enable the usage in various situations that can occur during processing:

- In the **Manual completion** mode, the operator manually triggers reading the values.
- In the **Automatic completion** mode, the phase reads the values and is completed automatically without any operator interaction.

As long as the phase is active, it provides user-triggered exceptions to override the retrieved values. If there are values affected by technical issues (red background color), the system lists those values at the top of the list of override exceptions to make them more easily locatable.

After completion the phase displays the affected properties and their values in the Execution Window.

The Navigator displays the identifier of the affected equipment entity.

Retrieve the process values from the tablet press.

Entity: TabletPress_AM / Automated Tablet Press

Property	Expected	Limits (LL L)	Value	Limits (H HH)	UoM	Timestamp
Compressing Force		65 ---	71	--- 75		04/14/2016 12:18:32 PM CEST
Tablet Dimensions		8.8 8.9	9.1	9.2 9.3 mm		04/14/2016 12:18:32 PM CEST
Tablet Form	Yes		Yes			04/14/2016 12:18:32 PM CEST
Batch ID (TP)	BX2		BX2			04/14/2016 12:18:32 PM CEST

Get

Confirm


Figure 46: Get OPC values during execution

Compressing Force

Override recorded value:

Current value 71

Override value


Confirm 

Tablet Dimensions

Override recorded value:

Current value 9.1 mm

Override value mm


Confirm 

Tablet Form

Override recorded value:

Current value Yes

Override value ☒ Yes ☐ No

Confirm 

Batch ID (TP)

Override recorded value:

Current value BX2

Override value


Confirm 

Figure 47: User-triggered exceptions of Get OPC values

Retrieve the process values from the tablet press.

Entity: TabletPress_AM / Automated Tablet Press

Property	Expected	Limits (LL L)	Value	Limits (H HH)	UoM	Timestamp
Compressing Force		65 ---	71	--- 75		04/14/2016 12:18:32 PM CEST
Tablet Dimensions		8.8 8.9	9.1	9.2 9.3 mm		04/14/2016 12:18:32 PM CEST
Tablet Form	Yes		Yes			04/14/2016 12:18:32 PM CEST
Batch ID (TP)	BX2		BX2			04/14/2016 12:18:32 PM CEST



Get  Confirm 

Figure 48: Get OPC values after phase completion

Run Tablet Press
Collect OPC Values

TabletPress_AM

Figure 49: Get OPC values in the Navigator

Phase Design

The characteristics of the **Get OPC values** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions. In the second row the phase displays the identifier and the short description of the entity to which it refers. The next rows display a tabular view of relevant data of each affected property and its values, spanning all columns. If there are values that have not been defined, such as limits for a Numeric value, the respective table cell displays N/A (if no limit is defined) or --- (if one of the two possible limits is not defined). Table cells that can never hold an entry, such as limits for Boolean or String values, show with a gray background. Once the property values have been read, the **Timestamp** column displays the timestamp when the value was written into the OPC tag. When an operator adds an exception, the phase displays an exception marker at the affected value. The center column of the bottom row provides the **Get** button, while the right column contains the **Confirm** button.

When the phase is completed it shows the same three-column, four-row layout with the data of the read property values listed in the table.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

In addition to the permanent process parameters that are always present, the **Get OPC values** phase provides value bundles as optional process parameters, which you can insert if required.

You can add process parameter bundles for up to 50 values of three different data types (**Boolean**, **Numeric**, **String**) to the **Get OPC values** phase.



ADDING PARAMETER BUNDLES

1. Click the **Add parameter bundle** button.
The system opens an option list that holds all data types available for the value.
2. Select the type.
The system opens the **Add <Data Type>** dialog to define the value's identifier.
3. Type an identifier and click the **OK** button.
The system adds all process parameters of the bundle to the list of parameters.



REMOVING PARAMETER BUNDLES

1. In the list of parameters, select header row that contains the identifier of the bundle you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the value bundle.

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Text	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).

Identified equipment entity

Indicates the equipment entity to be processed with the phase. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the output of a preceding phase that provides an identified equipment entity.

Mode

Defines if the phase expects operator interaction during execution.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. Manual completion (default): Operator confirms the phase. Automatic completion : Phase automatically sets the property values and is completed.

Override recorded value

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to override the value read from the equipment entity. It covers incidents when a communication issue causes a faulty value to be recorded, but the actual value is within the required range so that the process can be continued.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Boolean value - Master (bundle identifier)

Indicates the equipment property of the **Boolean** data type whose tags are accessed and read from the automation layer.

Attribute	Type	Comment
Property	String	Equipment property to be read.

For choosing an equipment property of a suitable data type, the system provides a Property Selection editor.

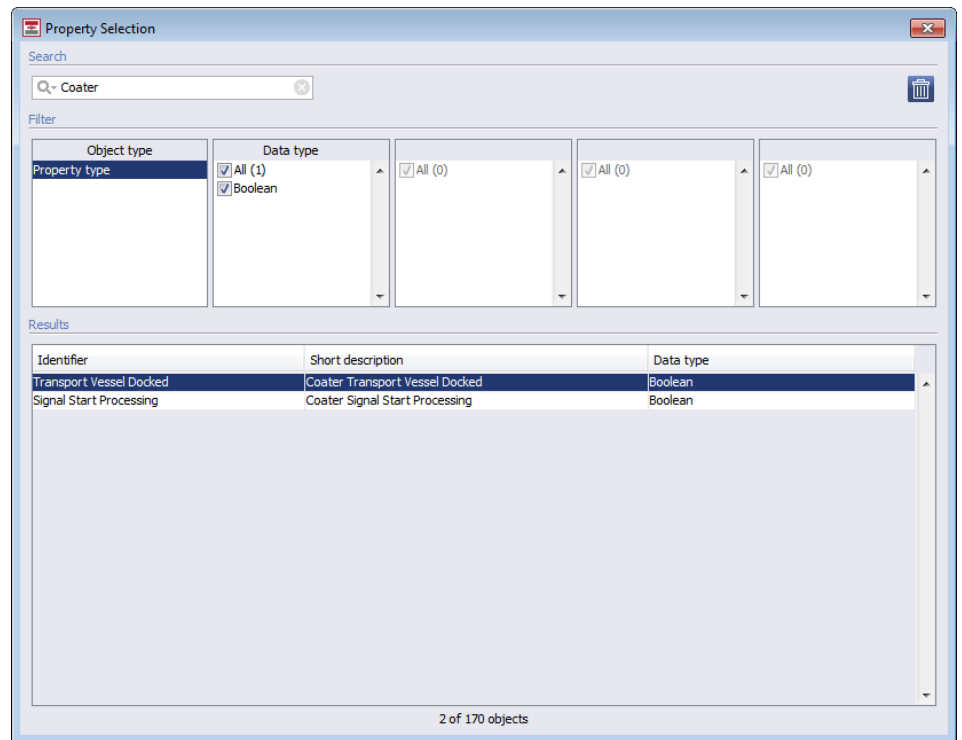


Figure 50: Property Selection editor for Automation properties (Boolean)

Expected value configuration

Defines if the value read from the automation layer must be checked against an expected value.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, make sure to set the Expected value attribute of the Expected value definition process parameter. If it is not set, the check fails. Default setting: No .
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .

Attribute	Type	Comment
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Expected value definition

Defines the value (by its display text) required as expected value if the respective check is enabled.

Attribute	Type	Comment
Value	Choice list	Defines the expected value. Available settings: N/A, Yes, No. Default setting: N/A.

Numeric value - Master (bundle identifier)

Indicates the equipment property of the **Numeric** data type whose tags are accessed and read from the automation layer.

Attribute	Type	Comment
Property	String	Equipment property to be read.

For choosing an equipment property of a suitable data type, the system provides a Property Selection editor.

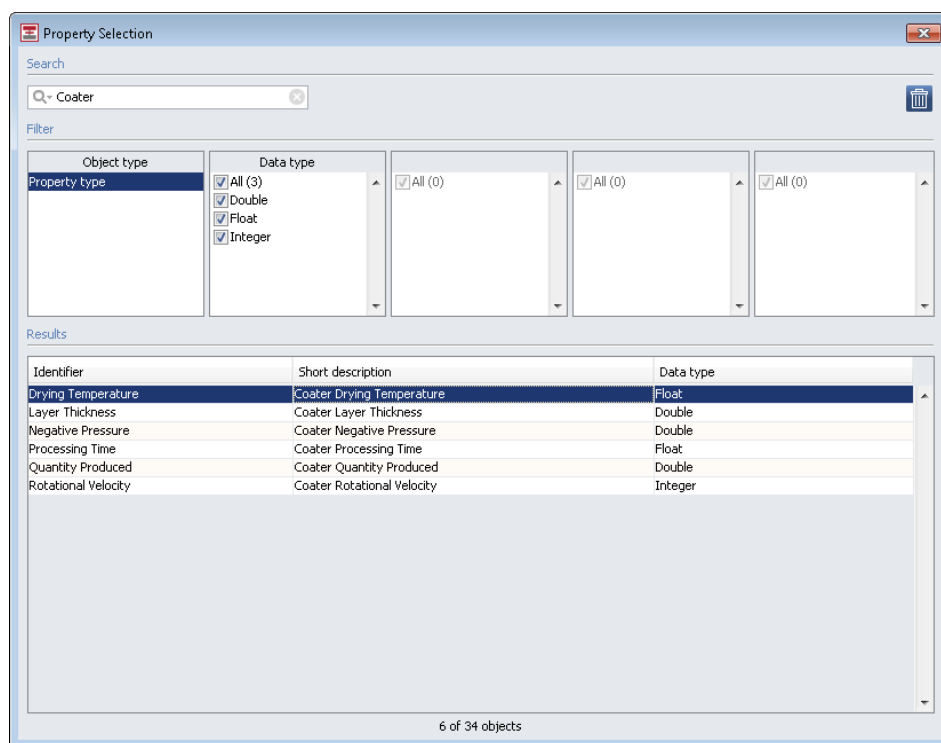


Figure 51: Property Selection editor for Automation properties (Numeric)

L-H configuration

Defines if the value read from the automation layer is checked against the limits defined with the **Limit definition** process parameter (page 79). If both checks are enabled, they are performed in the following order:

1. LL-HH
2. L-H

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, make sure to set at least one of the L limit or H limit attributes of the Limit definition process parameter (page 79). If they are not set, the check fails. Default setting: No .

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

LL-HH configuration

Defines if the value read from the automation layer is checked against the limits defined with the **Limit definition** process parameter (page 79). If both checks are enabled, they are performed in the following order:

1. LL-HH
2. L-H

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, make sure to set at least one of the LL limit or HH limit attributes of the Limit definition process parameter (page 79). If they are not set, the check fails. Default setting: No .
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Limit definition

Limits are defined as absolute values. The unit of measure is taken over from the equipment property. You can see the unit of measure in the Phase Preview.

Make sure that the limits are strictly sequential and do not overlap, so that

LL limit < L limit < H limit < HH limit

Attribute	Type	Comment
LL limit	BigDecimal (Double, Float, Integer)	Defines the values of the lower limits (including the values themselves). Limit values with more than 7 digits are truncated at the end in the Phase Preview.
L limit	BigDecimal (Double, Float, Integer)	
H limit	BigDecimal (Double, Float, Integer)	Defines the value of the upper limit (including the values themselves). Limit values with more than 7 digits are truncated at the end in the Phase Preview.
HH limit	BigDecimal (Double, Float, Integer)	

String value - Master (bundle identifier)

Indicates the equipment property of the **String** data type whose tags are accessed and written to the automation layer.

Attribute	Type	Comment
Property	String	Equipment property to be read.

For choosing an equipment property of a suitable data type, the system provides a Property Selection editor.

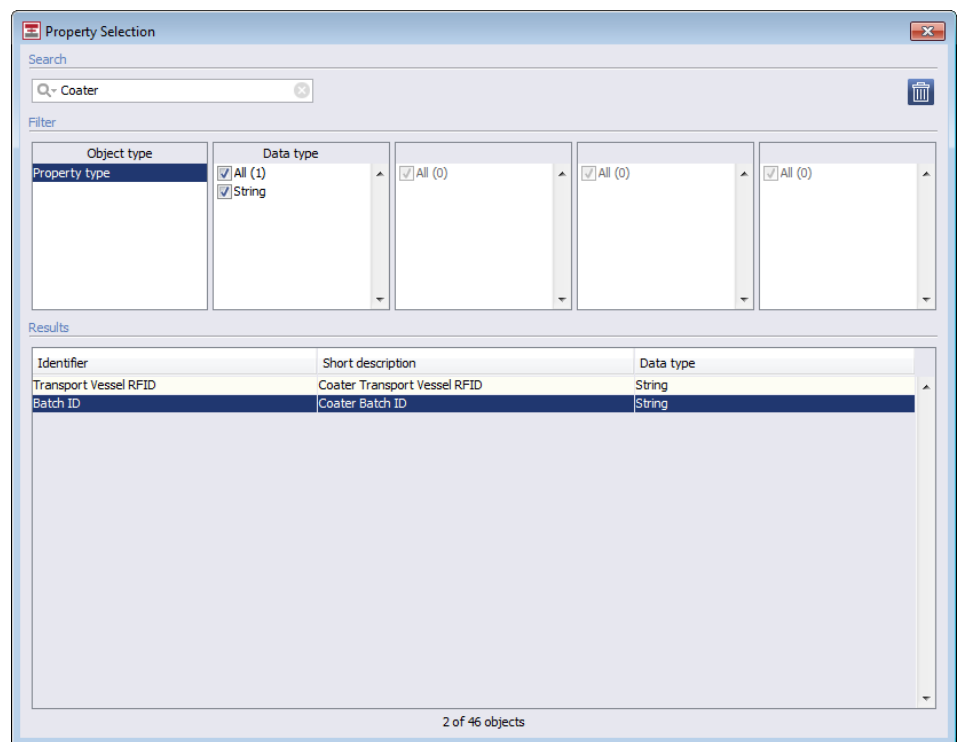


Figure 52: Property Selection editor for Automation properties (String)

Expected value configuration

Defines if the value read from the automation layer must be checked against an expected value.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, make sure to set the Expected value attribute of the Expected value definition process parameter. If it is not set, the check fails. Default setting: No .
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .

Attribute	Type	Comment
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Expected value definition

Defines the value required as expected value if the respective check is enabled.

Attribute	Type	Comment
Value	Text	Defines the expected value. Maximum length is 2000 characters.

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Set OPC values** phase provides the following output variables:

Automation set successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the automation layer was successful.
 - The value is `true` if all property values have been read successfully.
 - The value is `false` if at least one of the property values could not be read from the automation layer or has been overridden by using the respective user-triggered exception.

Boolean value

If you have added Boolean value bundles to the list of parameters, the system provides output variables for each of the bundles.

TIP

The output variables of a value bundle are prefixed with its bundle identifier.

The following output variables are available for Boolean value bundles:

Boolean value - Automation get successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the automation layer was successful.
 - The value is `true` if all property values have been read successfully.
 - The value is `false` if at least one of the property values could not be read from the automation layer or has been overridden by using the respective user-triggered exception.

Boolean value - Value

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable provides the value of the boolean property tag. The value is Null if N/A is the phase result.

Numeric value

If you have added Numeric value bundles to the list of parameters, the system provides output variables for each of the bundles.

TIP

The output variables of a value bundle are prefixed with its bundle identifier.

The following output variables are available for Numeric value bundles:

Numeric value - Automation get successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the automation layer was successful.
 - The value is `true` if all property values have been read successfully.
 - The value is `false` if at least one of the property values could not be read from the automation layer or has been overridden by using the respective user-triggered exception.

Numeric value - Unit of measure

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**mm**" or "**ea**".
- Usage: The output variable provides the unit of measure of the numeric property tag. The value is Null if N/A is the phase result.

Numeric Value - Value

- Data type: **BigDecimal**, floating point number that allows calculating with greater precision than **Float**.
- Usage: The output variable provides the actual value of the numeric property tag as a **BigDecimal** value. The value is Null if N/A is the phase result.

String value

If you have added string value bundles to the list of parameters, the system provides output variables for each of the bundles.

TIP

The output variables of a value bundle are prefixed with its bundle identifier.

The following output variables are available for string value bundles:

String value - Automation get successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the automation layer was successful.
 - The value is `true` if all property values have been read successfully.
 - The value is `false` if at least one of the property values could not be read from the automation layer or has been overridden by using the respective user-triggered exception.

String value - Value

- Data type: String, used for displaying a pre-defined sequence of characters, such as **"mm"** or **"ea"**.
- Usage: The output variable provides the value of the string property tag. The value is Null if N/A is the phase result.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as **"Read Instruction"**.
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString
(
  convertTo
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
    "min")
)
+ " min"
```

As result of the expression, the system displays "92 min".

-
-
- Rockwell Software PharmaSuite® BB - User Manual Equipment Phases
-
-

Set OPC Values

The **Set OPC values** phase allows to write up to 50 tag values of one equipment entity to the automation layer.

It can be used for processing requirements, such as:

- Set up parameters of a mixer
With one button tap, an operator can transfer the values of all relevant set points of a mixer to the automation layer:
 - Mixer speed = 500 (rpm), specification limit low = 400 rpm, specification limit high = 1000 rpm
 - HeatingControl = Yes
 - HeatingTargetTemp = 55 °C
 - HeatingProfile = 7
 - HMI_InstructionText1 = Control visual foam situation

The phase supports the following data types: Boolean, BigDecimal, and String.

Execution

The **Set OPC values** phase displays the identifier and short description of the equipment entity to which the phase refers. When the operator taps the **Set** button, the phase accesses the automation layer and writes the tag values provided for the specified properties. For each numeric property, it can process up to three values (**Low**, **Value**, **High**). Which of the values are actually expected to be written during execution is configured with the property data of the equipment entity.

During execution, if a value is affected by a technical issue, such as a communication failure, its cell background assumes a different color (red) and displays a marker symbol (X).

Different phase modes enable the usage in various situations that can occur during processing:

- In the **Manual completion** mode, the operator manually sets the values.
- In the **Automatic completion** mode, the phase sets the values and is completed automatically without any operator interaction.

As long as the phase is active, it provides user-triggered exceptions

- to confirm that values have been set directly at the equipment entity and not through PharmaSuite
- to override the values sent to the equipment entity.

If there are values affected by technical issues (red background color), the system lists those values at the top of the list of override exceptions to make them more easily locatable.


After completion the phase displays the affected properties and their values in the Execution Window.

The Navigator displays the identifier of the affected equipment entity.

Download the setpoints to the tablet press.

Entity: TabletPress_AM / Automated Tablet Press

Property	Low	Value	High	UoM	Automation set
Compressing Force	65	70	75		<input checked="" type="checkbox"/>
Tablet Dimensions	8.8	9	9.2 mm		<input checked="" type="checkbox"/>
Tablet Form		Yes			<input checked="" type="checkbox"/>
Batch ID (TP)		BX57			<input checked="" type="checkbox"/>

Set 



Confirm 


Figure 53: Set OPC values during execution

Values set directly at equipment. Confirm 

Compressing Force

Override recorded values:


Current values (low, value, high) 65 70 75

Override values (low, value, high) Confirm 

Tablet Dimensions

Override recorded values:


Current values (low, value, high) 8.8 9 9.2 mm

Override values (low, value, high) mm Confirm 

Tablet Form

Override recorded value:

Current value Yes

Override value ☒ Yes ☐ No Confirm 

Batch ID (TP)

Override recorded value:

Current value BX57


Override value Confirm 

Figure 54: User-triggered exceptions of Set OPC values

Download the setpoints to the tablet press.

Entity: TabletPress_AM / Automated Tablet Press

Property	Low	Value	High	UoM	Automation set
Compressing Force	65	70	75		<input checked="" type="checkbox"/>
Tablet Dimensions	8.8	9	9.2 mm		<input checked="" type="checkbox"/>
Tablet Form		Yes			<input checked="" type="checkbox"/>
Batch ID (TP)		BX57			<input checked="" type="checkbox"/>



Set  Confirm 

Figure 55: Set OPC values after phase completion

Run Tablet Press Download Setpoints TabletPress_AM

Figure 56: Set OPC values in the Navigator

Phase Design

The characteristics of the **Set OPC values** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions. In the second row the phase displays the identifier and the short description of the entity to which it refers. The next rows display a tabular view of relevant data of each affected property and its values, spanning all columns. If there are values that have not been defined with the equipment, such as **High** or **Low** for a Numeric value, the respective table cell displays N/A. Table cells that can never hold an entry, such as **High** or **Low** for Boolean or String values, show with a gray background. Once the property values have been written, the **Automation set** column displays a checkmark if all values have been set successfully. When an operator adds an exception, the phase displays an exception marker at the affected value. The center column of the bottom row provides the **Set** button, while the right column contains the **Confirm** button.

When the phase is completed it shows the same three-column, four-row layout with the data of the written property values listed in the table.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Text	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).

Identified equipment entity

Indicates the equipment entity to be processed with the phase. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the output of a preceding phase that provides an identified equipment entity.

Mode

Defines if the phase expects operator interaction during execution.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. Manual completion (default): Operator confirms the phase. Automatic completion : Phase automatically sets the property values and is completed.

Override value definition

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to override the value defined by the process parameter or the equipment entity.

It covers incidents when a property's **Source** attribute sets the boolean (page 93), numeric (page 94), or string (page 95) value to be drawn from the entity, but another entity than intended is identified so that the value drawn from the identified entity is not suitable for the process and needs to be corrected.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .

Attribute	Type	Comment
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Input at equipment

Represents a user-triggered exception that is accessible from the Exception Window. It allows an operator to document that property values have been set manually at a machine interface of the physical equipment entity. With the exception the operator confirms that the values have been set as documented by the phase.

It covers incidents when a communication issue prevents the system from accessing the equipment entity and the operator sets the values directly at the entity's machine interface to enable the process to continue.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Values

In addition to the permanent process parameters that are always present, the **Set OPC values** phase provides value bundles as optional process parameters, which you can insert if required.

You can add process parameter bundles for up to 50 values of three different data types (**Boolean**, **Numeric**, **String**) to the **Set OPC values** phase.



ADDING PARAMETER BUNDLES

1. Click the **Add parameter bundle** button.
The system opens an option list that holds all data types available for the value.
2. Select the type.
The system opens the **Add <Data Type>** dialog to define the value's identifier.
3. Type an identifier and click the **OK** button.
The system adds all process parameters of the bundle to the list of parameters.



REMOVING PARAMETER BUNDLES

1. In the list of parameters, select header row that contains the identifier of the bundle you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the value bundle.

The following process parameters are available to configure the phase's behavior during execution:

Boolean value

Indicates the equipment property of the **Boolean** data type whose tags are accessed and written to the automation layer.

Attribute	Type	Comment
Property	String	Equipment property to be written.
Value	Boolean	Value to be set.
Source	Choice list	Defines if the value definition is taken from the process parameter or the equipment entity. Default setting: Process parameter .

For choosing an equipment property of a suitable data type, the system provides a Property Selection editor.

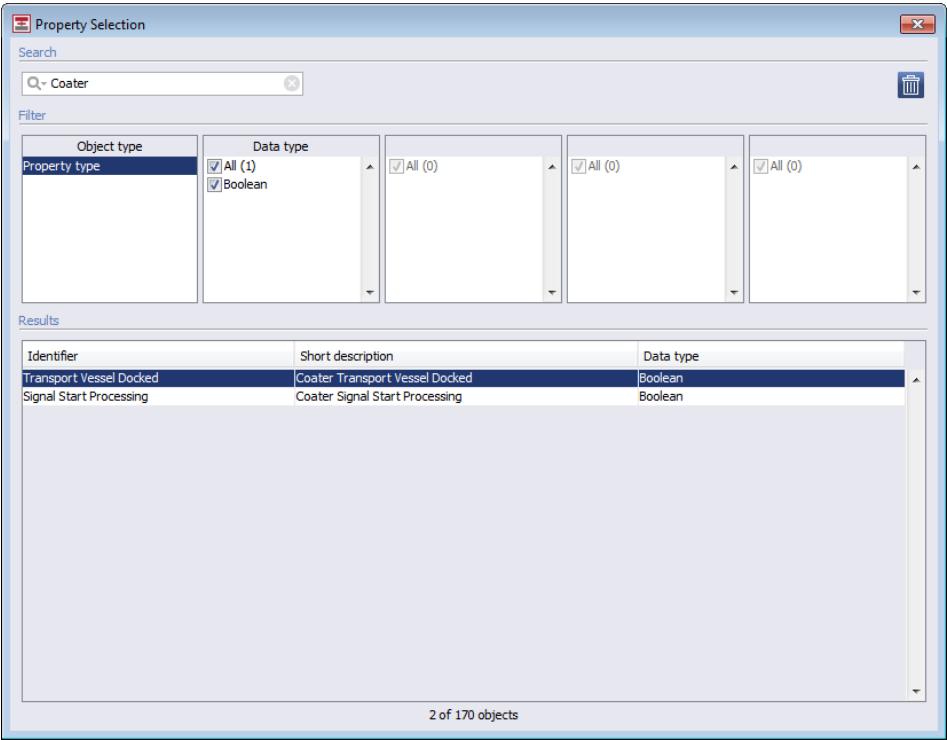


Figure 57: Property Selection editor for Automation properties (Boolean)

Numeric value

Indicates the equipment property of the **Numeric** data type whose tags are accessed and written to the automation layer.

Attribute	Type	Comment
Property	String	Equipment property to be written.
Low	BigDecimal (Double, Float, Integer)	Value to be set.
Value	BigDecimal (Double, Float, Integer)	Value to be set.
High	BigDecimal (Double, Float, Integer)	Value to be set.
Source	Choice list	Defines if the value definition is taken from the process parameter or the equipment entity. Default setting: Process parameter .

For choosing an equipment property of a suitable data type, the system provides a Property Selection editor.

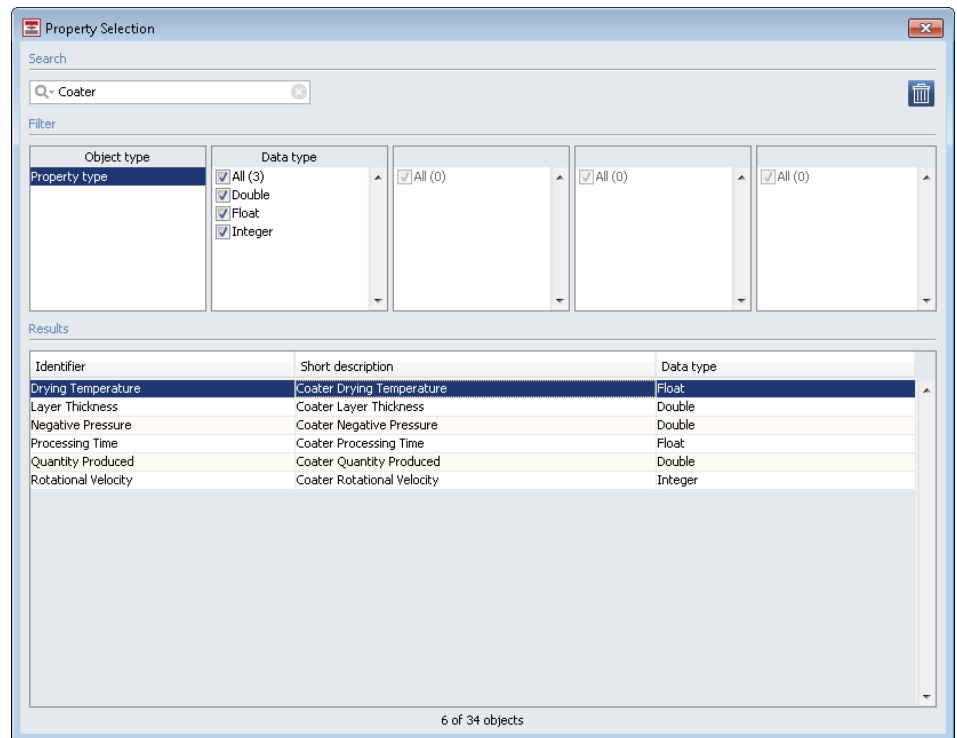


Figure 58: Property Selection editor for Automation properties (Numeric)

String value

Indicates the equipment property of the **String** data type whose tags are accessed and written to the automation layer.

Attribute	Type	Comment
Property	String	Equipment property to be written.
Value	String	Value to be set.
Source	Choice list	Defines if the value definition is taken from the process parameter or the equipment entity. Default setting: Process parameter .

For choosing an equipment property of a suitable data type, the system provides a Property Selection editor.

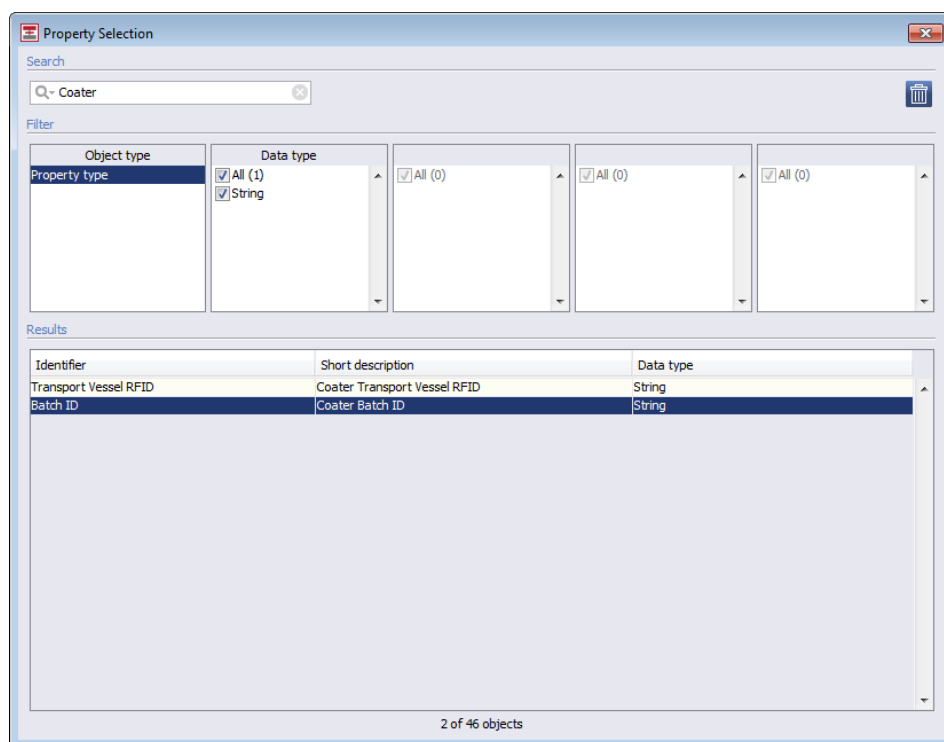


Figure 59: Property Selection editor for Automation properties (String)

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Set OPC values** phase provides the following output variables:

Automation set successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the set operation on the automation layer was successful.
 - The value is `true` if all property values have been set.
 - The value is `false` if at least one of the property values could not be set.

Boolean value

If you have added Boolean value bundles to the list of parameters, the system provides output variables for each of the bundles.

TIP

The output variables of a value bundle are prefixed with its bundle identifier.

The following output variables are available for Boolean value bundles:

Boolean value - Value

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable provides the value of the boolean property tag. The value is Null if N/A is the phase result.

Boolean value - Automation set successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the set operation on the automation layer was successful.
 - The value is `true` if all property values have been set.
 - The value is `false` if at least one of the property values could not be set.

Numeric value

If you have added Numeric value bundles to the list of parameters, the system provides output variables for each of the bundles.

TIP

The output variables of a value bundle are prefixed with its bundle identifier.

The following output variables are available for Numeric value bundles:

Numeric Value - Value

- Data type: BigDecimal, floating point number that allows calculating with greater precision than Float.
- Usage: The output variable provides the actual value of the numeric property tag as a **BigDecimal** value. The value is Null if N/A is the phase result.

Numeric value - Low

- Data type: BigDecimal, floating point number that allows calculating with greater precision than Float.
- Usage: The output variable provides the actual value of the numeric property tag as a **BigDecimal** value. The value is Null if N/A is the phase result.

Numeric value - High

- Data type: BigDecimal, floating point number that allows calculating with greater precision than Float.
- Usage: The output variable provides the actual value of the numeric property tag as a **BigDecimal** value. The value is Null if N/A is the phase result.

Numeric value - Unit of measure

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**mm**" or "**ea**".
- Usage: The output variable provides the unit of measure of the numeric property tag. The value is Null if N/A is the phase result.

Numeric value - Automation set successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the set operation on the automation layer was successful.
 - The value is `true` if all property values have been set.
 - The value is `false` if at least one of the property values could not be set.

String value

If you have added string value bundles to the list of parameters, the system provides output variables for each of the bundles.

TIP

The output variables of a value bundle are prefixed with its bundle identifier.

The following output variables are available for string value bundles:

String value - Value

- Data type: String, used for displaying a pre-defined sequence of characters, such as **"mm"** or **"ea"**.
- Usage: The output variable provides the value of the string property tag. The value is Null if N/A is the phase result.

String value - Automation set successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the set operation on the automation layer was successful.
 - The value is `true` if all property values have been set.
 - The value is `false` if at least one of the property values could not be set.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as **"Read Instruction"**.
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString
(
  convertTo
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
    "min")
  )
+ " min"
```

As result of the expression, the system displays "92 min".

Monitor Numeric Value

The **Monitor numeric value** phase reads a numeric value within a defined monitoring period and compares the value with a pre-defined condition.

It can be used for processing requirements, such as:

- Waiting for a specific numeric value to reach a certain value before processing can continue
An agitator needs several minutes to reach the speed set-point of 50 rpm. The phase evaluates the speed value every 5 seconds and if 50 rpm is reached within a pre-defined monitoring period, the agitator is ready for use.
- Assuring that a specific numeric value does not exceed a pre-defined limit
Within a given monitoring period of 30 minutes, the phase evaluates the temperature every 10 seconds. If the temperature exceeds 30 °C, an exception is recorded.

Execution

Different phase modes enable the usage in various situations that can occur during processing:

- In the **Manual completion** mode, the operator manually completes the phase.
- In the **Automatic completion** mode, under certain conditions, the phase is automatically completed without any operator interaction.

The **Monitor numeric value** phase displays the following data:

- the identifier and short description of the equipment entity to which the phase refers
- the monitoring condition
- the monitoring duration
- the monitoring result
- the value that constitutes the monitoring result.

When the phase becomes active, it accesses the automation layer, reads the monitoring value, compares it against the condition, and calculates and displays the end time of the monitoring period. Monitoring continues until either the condition has been met or the monitoring period is over. While monitoring is ongoing the system displays an animated phase activity marker in its top right corner.

The update rate at which the monitoring tag is read is configured with the equipment entity in Data Manager.

TIPS

- Changes of a value that fulfill the condition only temporarily between two read-cycles are not detected.
- Due to technical reasons, the accuracy of the **Double** or **Float** numeric data type cannot be guaranteed. Hence, the result of the **Value == 13.4** condition can be unexpected if the value is of the **Double** or **Float** numeric data type.
- An Automation Integration server supports simultaneous monitoring calls from one or several clients, but only when they call different tags. Thus you cannot monitor the one and the same tag for different values.

Different phase modes enable the usage in various situations that can occur during processing:

- In the **Manual completion** mode, the operator manually completes the phase.
- In the **Automatic completion** mode, under certain conditions, the phase is automatically completed without any operator interaction.

As long as the phase is active, it provides a user-triggered exception to terminate monitoring before the monitoring period has expired and to record the result value manually.

After completion the phase displays the affected numeric property and its values in the Execution Window.

The Navigator displays the identifier of the affected equipment entity.



Figure 60: Monitor numeric value during execution



Figure 61: User-triggered exception of Monitor numeric value

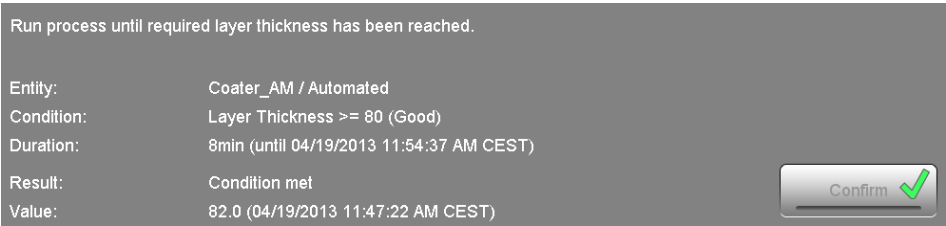


Figure 62: Monitor numeric value after phase completion



Figure 63: Monitor numeric value in the Navigator

Phase Design

The characteristics of the **Monitor numeric value** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span several rows. When the phase is active, the left and center columns of the first row provide space for textual instructions. The right column holds the animated phase activity marker. In the second row the phase displays the identifier and the short description of the entity to which it refers. The following rows list the monitoring data (**Condition**, **Duration**, **Result**, and **Value**). Once the monitoring data is available, it is displayed in the respective rows. The rightmost column provides the **Confirm** button.

When the phase is completed it shows the same three-column, six-row layout with the monitoring data filled in.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Not used.
Column 3	HTML text	Not used.

Identified equipment entity

Indicates the equipment entity to be processed with the phase. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the output of a preceding phase that provides an identified equipment entity.

Mode

Defines if the phase expects operator interaction during execution.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. Manual completion (default): Operator confirms the phase. Automatic completion : Phase automatically gets the property value and is completed.

Numeric property

Indicates the equipment property of the **Numeric** data type whose tag is accessed and read from the automation layer.

Attribute	Type	Comment
Property	String	Equipment property to be read.
Comparator	Choice list	Defines the comparison, where x is the monitored value. x == value1 : equal to x != value1 : not equal to x < value1 : less than x <= value 1 : less than or equal to x >= value1 : greater than or equal to x > value1 : greater than value1 <= x <= value2 : closed interval value1 < x < value2 : open interval Default setting: x >= value1
Value1	BigDecimal	Defines the first value of the comparison.
Value2	BigDecimal	Defines the second value of the comparison, if applicable.
Meaning of "condition met"	Choice list	Defines the string to be displayed as result of the comparison. Available settings: Good , Exception . Default setting: Good
Monitor duration	Duration	Defines the monitoring period in hh:mm:ss. The minimum duration is one second. Specified milliseconds are not displayed in the phase user interface. Note: The duration must always be longer than the tag update rate that is configured on equipment property level.

For choosing an equipment property of a suitable data type, the system provides a Property Selection editor.

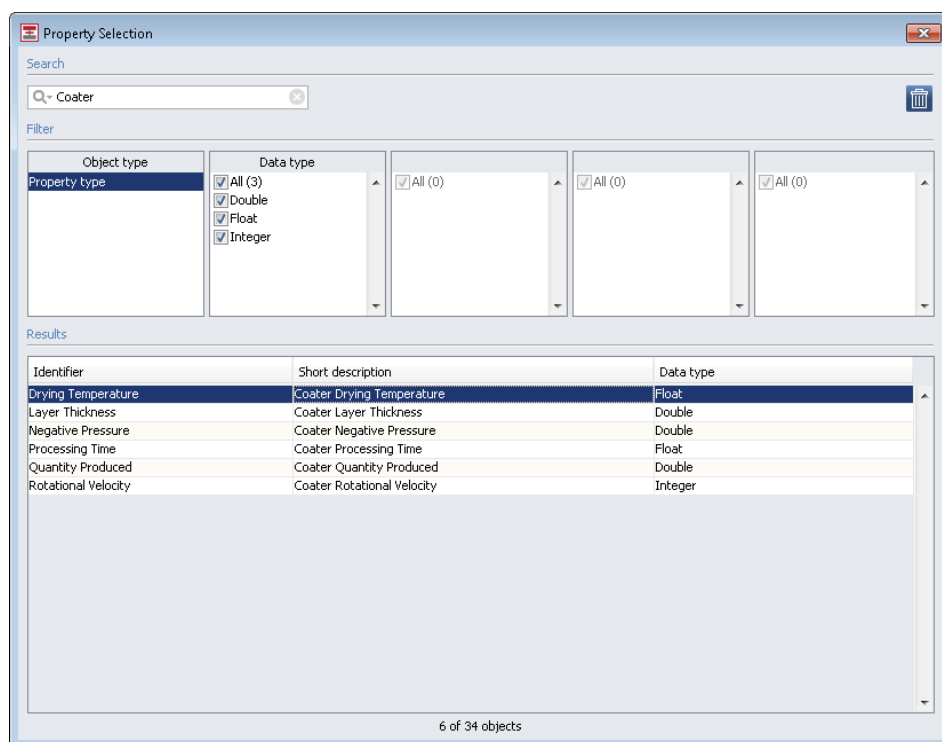


Figure 64: Property Selection editor for Automation properties (Numeric)

Monitoring exception

Represents a system-triggered exception that is displayed in the Exception Window to resolve incidents when either the condition is fulfilled and defined as **Exception** or the condition is defined as **Good** and the monitor duration has expired. The behavior of the exception is controlled by the **Comparator**, **Meaning of "condition met"**, and **Monitor duration** attributes of the **Numeric property** process parameter (page 105).

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Stop monitoring and record result

Represents a user-triggered exception that is accessible from the Exception Window. It allows an operator to terminate monitoring before the monitoring period has expired and to record the result value manually.

It covers incidents when a technical issue prevents the system from reading the value correctly, but a gage at the equipment entity itself shows that the condition has been met.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Unforeseen resume

Represents a system-triggered exception that is displayed in the Exception Window to resolve incidents when the system has experienced an unexpected shutdown and is resumed afterwards. In the batch record, the system automatically includes a general statement of the incident, to which you can add more specific instructions by means of the **Exception text** attribute.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Monitor numeric value** phase provides the following output variables:

Value

- Data type: **MeasuredValue**, used for displaying numeric values qualified by a unit of measure.
- Usage: The output variable provides the complete process value as a **MeasuredValue** object.

Timestamp of tag

- Data type: **Timestamp**, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the timestamp when the value was read.

Monitoring exception occurred

- Data type: **Boolean**, with the values **true** and **false**
- Usage: The output variable states if an exception has occurred while the phase was active.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString  
  (convertTo  
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},  
    "min")  
  )  
  + " min"
```

As result of the expression, the system displays **"92 min"**.

Get Alarms

The **Get alarms** phase polls alarm tags within a defined interval of a single equipment entity on the automation layer.

It can be used for processing requirements, such as:

- Recording alarms and follow-up actions
The status of alarm tags is polled every five seconds. In case an alarm has occurred, the alarm itself is documented and, according to the recipe design, related follow-up actions can be executed before the **Get alarms** phase is activated again.

The phase supports up to 20 alarm tags.

The tag quality is not evaluated while determining if an alarm or error condition has been reached.

Execution

The **Get alarms** phase displays the following data:

- the identifier and short description of the equipment entity to which the phase refers
- the identifier and short description of the bundle of properties collected in the property of the FlexibleTagDefinition type that is assigned as **Alarm property** parameter
- the overall status of the property bundle that is being observed
- the tag reading status, consisting of update rate and timestamp
- a tabular view listing the individual properties that constitute the property bundle.

When the phase becomes active, it polls the alarm tags for their statuses and displays small icons to indicate the status of each of the individual properties and a large icon to indicate the overall status of the property bundle.

There are three statuses indicated by different icons:



The checkmark icon indicates that the status of the alarm tag is as intended.



The alarm icon indicates that an alarm event has occurred at the alarm tag.



The error icon indicates that an error has occurred while reading the alarm tag.

As soon one of the monitored alarm tags returns an alarm or an error, the icon that indicated the overall status changes to reflect this. Polling continues until either an alarm occurs, the operator confirms the phase manually, or the phase completes automatically.

Different phase modes enable the usage in various situations that can occur during processing:

- In the **Manual completion** mode, the operator manually completes the phase.
- In the **Automatic completion** mode, under certain conditions, the phase is automatically completed without any operator interaction.

After completion the phase displays the affected alarm property, its tags, and their statuses in the Execution Window.

The Navigator displays the identifier of the affected equipment entity.

Monitor the listed values.

Entity: Coater_AM / Automated

Property: Monitor Coater / Monitor Coater

Status: Update every: 2s / Next update: 4/19/2013 11:59:48 AM CEST

✓ StatusValve1	✓ StatusValve2	✓ StatusValve3	✓ StatusValve4
✓ StatusValve5	✓ FrontDoor	✓ Enclosure	✓ AirPressure
✓ Heater	✓ DrumMotor		

Overall status: 

Confirm 


Figure 65: Get alarms during execution











Monitor the listed values.

Entity: Coater_AM / Automated

Property: Monitor Coater / Monitor Coater

Status: Stopped at: 4/19/2013 12:02:53 PM CEST

Overall status 

 StatusValve1	 StatusValve2	 StatusValve3	 StatusValve4
 StatusValve5	 FrontDoor	 Enclosure	 AirPressure
 Heater	 DrumMotor		

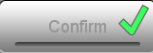


Figure 66: Get alarms after phase completion

Coating Process Monitoring
Monitor Coating Unit

Coater_AM

Figure 67: Get alarms in the Navigator

Phase Design

The characteristics of the **Get alarms** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions. In the second row the phase displays the identifier and the short description of the entity to which it refers. The third row holds the identifier and the short description of the FlexibleTagDefinition property and the fourth indicates the status of the tag reading. The right column provides space for displaying the large icon to indicate the overall status of the monitored bundle of properties.

The next row displays a table that spans all columns. Each cell of the table holds one of the monitored properties, preceded by its status icon.

The right column of the bottom row contains the **Confirm** button.

When the phase is completed it shows the same three-column, five-row layout with the status icons of the last read statuses listed in the table.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Not used.
Column 3	HTML text	Not used.

Alarm property

Indicates the equipment property of the **FlexibleTagDefinition** data type whose tags are accessed and read from the automation layer.

Attribute	Type	Comment
Property	String	Equipment property to be read. Supported property data type: FlexibleTagDefinition with tags of boolean Live Data type.
Update interval	Duration	Defines the interval in hh:mm:ss between read operations. The minimum interval is set to 1 second if the interval is not defined at all or configured to be less than that.

For choosing an equipment property of a suitable data type, the system provides a Property Selection editor.

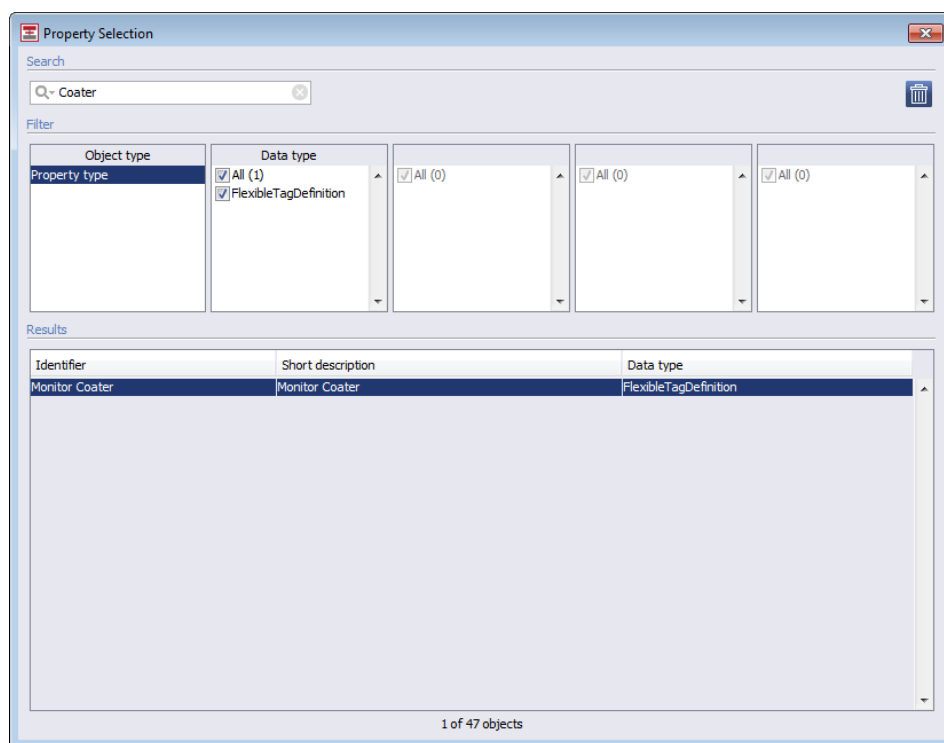


Figure 68: Property Selection editor for Automation properties (FlexibleTagDefinition)

Identified equipment entity

Indicates the equipment entity to be processed with the phase. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the output of a preceding phase that provides an identified equipment entity.

Mode

Defines if the phase expects operator interaction during execution.

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. Manual completion (default): Operator confirms the phase. Automatic completion : Phase automatically gets the property values and is completed.

Alarm exception

Represents a system-triggered exception that is displayed in the Exception Window to resolve incidents when an alarm is raised or an error has occurred.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, the phase creates a system-triggered exception when an alarm is raised. Default setting: Yes .
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Unforeseen resume

Represents a system-triggered exception that is displayed in the Exception Window to resolve incidents when the system has experienced an unexpected shutdown and is resumed afterwards. In the batch record, the system automatically includes a general statement of the incident, to which you can add more specific instructions by means of the **Exception text** attribute.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .

Attribute	Type	Comment
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Get alarms** phase provides the following output variables:

Alarm tags

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Coater-A;Coater-AL;Blender-S**".
- Usage: The output variable provides a semicolon-separated list of the tag identifiers for which an alarm has been set. The list is empty if no alarm has occurred.

Overall status

- Data type: String, used for displaying a pre-defined sequence of characters, such as **"GOOD"** or **"PLANNED"**.
- Usage: The output variable provides the overall status of the alarm tags.
 - The value is `GOOD` if no alarm has been raised, no error has occurred, and all tags have been read successfully.
 - The value is `ALARM` if at least one alarm has been raised, regardless of the statuses of all other tags.
 - The value is `ERROR` if at least one error has occurred and no alarms have been raised, regardless of whether all other alarm tags have been read successfully.

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as **"Read Instruction"**.
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString
(
  convertTo
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},
    "min")
)
+ " min"
```

As result of the expression, the system displays "92 min".

-
-
- Rockwell Software PharmaSuite® BB - User Manual Equipment Phases
-
-

Show Historical Data Chart

The **Show historical data chart** phase allows to show a time series chart for historical data.

It can be used for processing requirements, such as:

- History of a full process run
Display an overview of a full process run with various process parameters. This can be used as a trigger for further analysis with other tools if required.
- History of a specific timeframe
Display a detailed process view of a specific timeframe with one or more process parameters to see if unexpected or irregular values have occurred.

Execution

The **Show historical data chart** phase displays the identifier and short description of the equipment entity to which the phase refers and loads the chart. If a temporary technical issue has caused the loading of the chart to fail, the operator can request to reload the chart.

The chart displays all plots and y-axes as defined in the phase's process parameters (page 123). The x-axis shows the time scale for the entire period considered for the chart. Gaps in a plot thus indicate that during that time the Historian application did not record any values.

Below the chart, a legend indicates the meaning of the chart by showing the individual plot lines together with the short description and, if applicable, the unit of measure of the property to which they refer. This means that the legend texts are maintained with the data of the respective property types in Data Manager.

As long as the phase is active, it provides a user-triggered exception to confirm the phase even though the chart is not available for display.

After completion the phase displays the identifier and short description of the affected equipment entity as well as the chart.

The Navigator displays the identifier of the affected equipment entity.

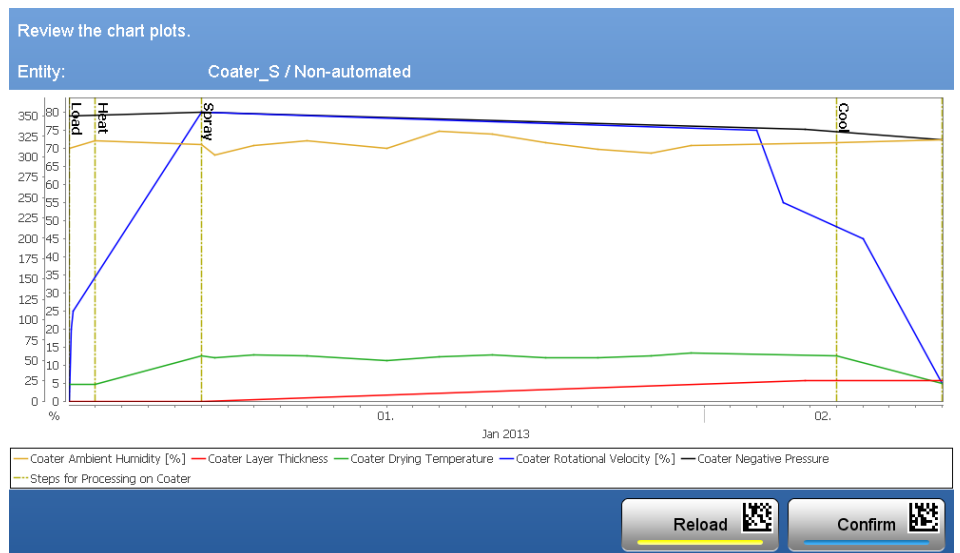


Figure 69: Show historical data chart during execution

Confirm without chart.

Confirm

Add exception

Risk Status Open

Performed by Password

Add comment

Figure 70: User-triggered exception of Show historical data chart

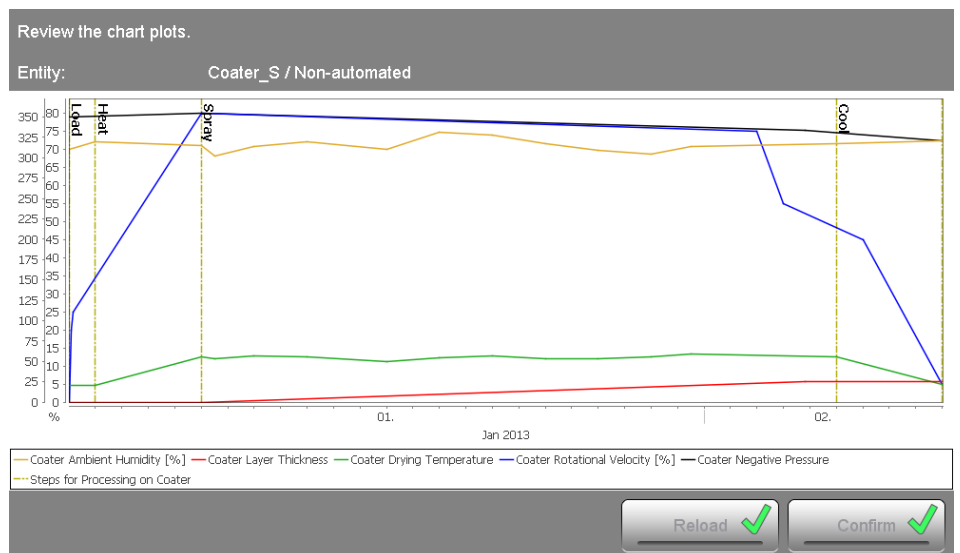


Figure 71: Show historical data chart after phase completion

Manual Coating Review Data Chart

Coater_S

Figure 72: Show historical data chart in the Navigator

Phase Design

The characteristics of the **Show historical data chart** phase are defined via process parameters and their attributes.

Its user interface is designed in three columns that span several rows. When the phase is active, the merged columns of the first row provide space for textual instructions. In the second row the phase displays the identifier and the short description of the entity to which it refers. The next row displays the chart image rendered according to the data received from the Historian server, spanning all columns. The center column of the bottom row provides the **Reload** button, while the right column contains the **Confirm** button.

When the phase is completed it shows the same three-column, four-row layout with the confirmed image of the chart in the third row.

Exception handling during execution is controlled by a risk assessment classification and an exception message that are both defined by the recipe author in the exception's process parameter.

Process Parameters

The following process parameters are available to configure the phase's behavior during execution:

Instruction

Represents the instruction text that is visible on the preview, the active, and the completed view of the phase.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	Not used.
Column 3	HTML text	Not used.

Identified equipment entity

Indicates the equipment entity to be processed with the phase. You cannot define a specific equipment identifier in the box, but have to use the Expression editor to select the equipment object output of a preceding **Identify equipment** phase (page 30).

Attribute	Type	Comment
Equipment object	Reference	Reference to the output of a preceding phase that provides an identified equipment entity.

Chart plot 1 - Chart plot 16

Indicate the properties shown as individual plot lines whose tags are accessed and read from the Historian server. For each plot line you can select a query template that specifies which data is retrieved and you can choose the color and format of the line itself. When you define your plots you need to keep in mind that the y-axes of your chart represent the units of measure for your plots. Since **Chart axis** parameters (page 126) only provide up to four y-axes, your plots can only have a maximum of four different units of measure. Plots that share a unit of measure are displayed on a y-axis that combines all ranges of the affected plots.

Attribute	Type	Comment
Enabled	Flag	Controls if the plot configuration is used to draw a plot.
Property	String	Historian point to be read.
Query template	Choice list	Defines the template to be used for data retrieval. By default, the system provides three templates: Raw archive data provides all archived data for the history period. Plot data provides the data dedicated to plotting (trending) applications within the history period. Marker (DigitalState) provides the string values within the history period that are available for use as segment markers.

Attribute	Type	Comment
Plot format	Choice list	Defines the plot drawing to be used with regard to line color, thickness, and line pattern. Dashed or dotted formatting should only be used for vertical markers, retrieved with the Marker (DigitalState) query template.
Timestamp 1	Timestamp	Optional parameter to be passed to the query template. In the default query templates, it indicates the start date and time of the chart period.
Timestamp 2	Timestamp	Optional parameter to be passed to the query template. In the default query templates, it indicates the end date and time of the chart period.
String 1	String	Optional parameter to be passed to the query template.
String 2	String	Optional parameter to be passed to the query template.
Duration	Duration	Optional parameter to be passed to the query template.
Long	Long	Optional parameter to be passed to the query template.
MeasuredValue 1	MeasuredValue	Optional parameter to be passed to the query template.
MeasuredValue 2	MeasuredValue	Optional parameter to be passed to the query template.

For choosing an equipment property of a suitable data type, the system provides a Property Selection editor.

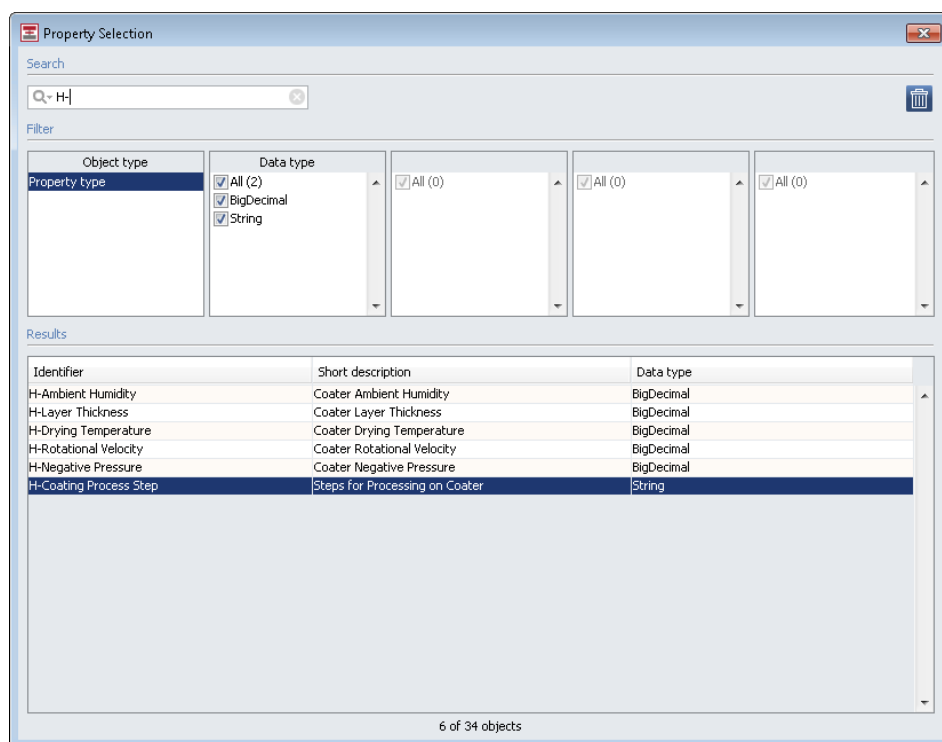


Figure 73: Property Selection editor for Historian properties

Chart axis 1 - Chart axis 4

Define up to four different y-axes to be displayed in the chart. The number of axes you need to define depends on the number of units of measure required by the plots you wish to display and specify in the **Chart plot** parameters (page 124). This means you can only have a maximum of four different units of measure for your plots. Plots of the same unit of measure are covered by one axis, which consequently needs to be defined with a range to accommodate all plots to which it pertains. It is also possible to define one unitless axis.

A plot using a template to display a vertical marker does not need to have a specific axis

Attribute	Type	Comment
Enabled	Flag	Controls if the axis configuration is used to draw an axis.
Unit of measure	Choice list	Defines the unit of measure for the axis and thus its label.
Minimum	Numeric	The minimum value shown on the y-axis. If one of the plots to which the y-axis refers has values below the minimum, they are not drawn on the chart. A value is required if Autorange is disabled.

Attribute	Type	Comment
Maximum	Numeric	The maximum value shown on the y-axis. If one of the plots to which the y-axis refers has values above the maximum, they are not drawn on the chart. A value is required if Autorange is disabled.
Autorange	Flag	Defines if the minimum and maximum values of the y-axis are to be determined automatically. If so, the values defined for the Minimum and Maximum attributes are ignored. Default setting: Yes
Scaling mode	Choice list	Defines the scaling mode. Available settings: Linear , Logarithmic . Default setting: Linear .
Number format	String	Defines the format pattern for numeric values. Examples: 0 : Integer portion of a number. 0.0 : Integer portion and one fractional digit. 0.00 : Integer portion and two fractional digits. Default setting: 0 .

TIP

Chart rendering will fail if negative values are to be rendered on a logarithmic scale.

Chart unavailable

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to confirm the phase even though the chart image cannot be loaded.

It covers incidents when a communication issue causes the chart image to be unavailable, but the operator could confirm by other means that the chart indicates no irregularities, so that the process can be continued.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: None , Low , Low (mandatory comment) , Medium , Medium (mandatory comment) , High , High (mandatory comment) . Default setting: High .
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

Output Variables

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

- Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.
- Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Show historical data chart** phase provides the following output variables:

Chart available

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the chart is available (**TRUE**) or not (**FALSE**).

Identifier

- Data type: String, used for displaying a pre-defined sequence of characters, such as "**Read Instruction**".
- Usage: The output variable provides the identifier of the phase.

Instance count

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the count of the number of instances the phase has been processed, for example in a loop. The count is also increased when the phase is skipped from an operator's perspective, since the phase is still executed, but as a hidden phase.
The count variable of a phase that has not been executed provides 0 as output value.

Start time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the start time of the phase.

Completion time

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
To use a timestamp in a phase attribute, you have to make sure it has a matching data type, so to display it in an instruction text, you have to convert it into a string.
- Usage: The output variable provides the completion time of the phase.

TIP

To calculate a duration from two timestamps and display it in a specific format, you need to use two conversion functions on the calculation:

- **Convert to Unitless Number (convertTo)** takes the calculated duration and converts it into the duration's value for one of its units (e.g. minutes or seconds).
- **Convert to String for Display (convertToDisplayString)** takes the converted value and displays it as string to which you can add the unit, also as string.

Example:

Sample Phase with Start time = 14-Nov-2014@10:15

Sample Phase with Completion time = 14-Nov-2014@11:47

The duration is to be displayed in minutes.

```
convertToDisplayString  
  (convertTo  
    ({Sample Phase}.{Completion time}-{Sample Phase}.{Start time},  
    "min")  
  )  
  + " min"
```

As result of the expression, the system displays **"92 min"**.

A

- Alarm exception (Get alarms, parameter) • 116
- Alarm property (Get alarms, parameter) • 114
- Alarm tags (Get alarms, output) • 117
- Allowed change actions (Change equipment status, parameter) • 36
- Allowed triggers (Trigger graph transition, parameter) • 45
- Automation get successful (Get OPC values, output) • 82
 - Boolean value bundle • 82
 - Numeric value bundle • 83
 - String value bundle • 84
- Automation set successful (Set OPC values, output) • 97
 - Boolean value bundle • 97
 - Numeric value bundle • 98
 - String value bundle • 99

B

- Boolean value bundle (output)
 - Get OPC values • 82
 - Set OPC values • 97
- Boolean value bundle (parameter)
 - Get OPC values • 74
 - Set OPC values • 93

C

- Change equipment status • 33
 - Output variable • 38
 - Process parameter • 35
- Chart available (Show historical data chart, output) • 129
- Chart axis (Show historical data chart) • 126
- Chart plot (Show historical data chart, parameter) • 124
- Chart unavailable (Show historical data chart) • 128
- Child entity to be separated (Separate equipment, parameter) • 56
- Completion time (output)
 - Change equipment status • 39
 - Get alarms • 119

- Get OPC values • 85
- Identify equipment • 31
- Monitor numeric value • 109
- Separate equipment • 59
- Set OPC values • 100
- Show historical data chart • 129
- Trigger graph transition • 49
- Unbind equipment • 67
- Conventions (typographical) • 2

E

- Enter identifier manually (Identify equipment, parameter) • 28
- Equipment data mismatch (parameter)
 - Change equipment status • 37
 - Trigger graph transition • 46
 - Unbind equipment • 65
- Equipment ID (Identify equipment, output) • 30
- Equipment object (Identify equipment, output) • 30
- Equipment short description (Identify equipment, output) • 30
- Equipment status check (Identify equipment, parameter) • 27
- Equipment unbind failed (Unbind equipment, parameter) • 65
- Expected value configuration (Get OPC values, parameter)
 - Boolean value bundle • 75
 - String value bundle • 80
- Expected value definition (Get OPC values, parameter)
 - Boolean value bundle • 76
 - String value bundle • 81

F

- Force status (Change equipment status, parameter) • 37

G

- Get alarms • 111
 - Output variable • 117

Process parameter • 114
Get OPC values • 69
Output variable • 81
Process parameter • 72
Graph (Trigger graph transition, output) • 48

H

High - Numeric value bundle (Set OPC values, output) • 98

I

Identification mode (Identify equipment, parameter) • 25
Identified equipment entity (parameter)
Change equipment status • 35
Get alarms • 115
Get OPC values • 73
Monitor numeric value • 104
Set OPC values • 91
Show historical data chart • 124
Trigger graph transition • 45
Unbind equipment • 63

Identifier (output)

Change equipment status • 38
Get alarms • 118
Get OPC values • 84
Identify equipment • 30
Monitor numeric value • 109
Separate equipment • 58
Set OPC values • 99
Show historical data chart • 129
Trigger graph transition • 49
Unbind equipment • 66

Identify equipment • 19

Output variable • 29
Process parameter • 24

Identify in current binding context (Identify equipment, parameter) • 25

Input at equipment (Set OPC values, parameter) • 92

Instance count (output)

Change equipment status • 39
Get alarms • 118
Get OPC values • 84
Identify equipment • 31

Monitor numeric value • 109
Separate equipment • 58
Set OPC values • 99
Show historical data chart • 129
Trigger graph transition • 49
Unbind equipment • 66

Instruction (parameter)

Change equipment status • 35
Get alarms • 114
Get OPC values • 73
Identify equipment • 24
Monitor numeric value • 104
Separate equipment • 55
Set OPC values • 90
Show historical data chart • 123
Trigger graph transition • 44
Unbind equipment • 63

K

Keep binding (Separate equipment, parameter) • 57
Keep sub-group (Separate equipment, parameter) • 56

L

L-H configuration - Numeric value bundle (Get OPC values, parameter) • 77
Limit definition - Numeric value bundle (Get OPC values, parameter) • 79
LL-HH configuration - Numeric value bundle (Get OPC values, parameter) • 78
Low - Numeric value bundle (Set OPC values, output) • 98

M

Mode (parameter)

Get alarms • 115
Get OPC values • 73
Identify equipment • 24
Monitor numeric value • 104
Separate equipment • 55
Set OPC values • 91
Trigger graph transition • 44
Unbind equipment • 64
Monitor numeric value • 101

Output variable • 108

Process parameter • 104

Monitoring exception (Monitor numeric value, parameter)
• 106

Monitoring exception occurred (Monitor numeric value,
output) • 108

N

New status (output)

Change equipment status • 38

Trigger graph transition • 48

Numeric property (Monitor numeric value, parameter) •
105

Numeric value bundle (output)

Get OPC values • 83

Set OPC values • 97

Numeric value bundle (parameter)

Get OPC values • 76

Set OPC values • 94

O

Old status (Trigger graph transition, output) • 48

Output variable

Alarm tags (Get alarms) • 117

Automation get successful - Boolean value bundle (Get
OPC values) • 82

Automation get successful - Numeric value bundle (Get
OPC values) • 83

Automation get successful - String value bundle (Get
OPC values) • 84

Automation get successful (Get OPC values) • 82

Automation set successful - Boolean value bundle (Set
OPC values) • 97

Automation set successful - Numeric value bundle (Set
OPC values) • 98

Automation set successful - String value bundle (Set
OPC values) • 99

Automation set successful (Set OPC values) • 97

Boolean value bundle (Get OPC values) • 82

Boolean value bundle (Set OPC values) • 97

Change equipment status • 38

Chart available (Show historical data chart) • 129

Completion time (Change equipment status) • 39

Completion time (Get alarms) • 119

Completion time (Get OPC values) • 85

Completion time (Identify equipment) • 31

Completion time (Monitor numeric value) • 109

Completion time (Separate equipment) • 59

Completion time (Set OPC values) • 100

Completion time (Show historical data chart) • 129

Completion time (Trigger graph transition) • 49

Completion time (Unbind equipment) • 67

Equipment ID (Identify equipment) • 30

Equipment object (Identify equipment) • 30

Equipment short description (Identify equipment) • 30

Get alarms • 117

Get OPC values • 81

Graph (Trigger graph transition) • 48

High - Numeric value bundle (Set OPC values) • 98

Identifier (Change equipment status) • 38

Identifier (Get alarms) • 118

Identifier (Get OPC values) • 84

Identifier (Identify equipment) • 30

Identifier (Monitor numeric value) • 109

Identifier (Separate equipment) • 58

Identifier (Set OPC values) • 99

Identifier (Show historical data chart) • 129

Identifier (Trigger graph transition) • 49

Identifier (Unbind equipment) • 66

Identify equipment • 29

Instance count (Change equipment status) • 39

Instance count (Get alarms) • 118

Instance count (Get OPC values) • 84

Instance count (Identify equipment) • 31

Instance count (Monitor numeric value) • 109

Instance count (Separate equipment) • 58

Instance count (Set OPC values) • 99

Instance count (Show historical data chart) • 129

Instance count (Trigger graph transition) • 49

Instance count (Unbind equipment) • 66

Low - Numeric value bundle (Set OPC values) • 98

Monitor numeric value • 108

- Monitoring exception occurred (Monitor numeric value) • 108
- New status (Change equipment status) • 38
- New status (Trigger graph transition) • 48
- Numeric value bundle (Get OPC values) • 83
- Numeric value bundle (Set OPC values) • 97
- Old status (Trigger graph transition) • 48
- Overall status (Get alarms) • 118
- Result (Identify equipment) • 30
- Selected trigger (Trigger graph transition) • 48
- Separate equipment • 58
- Set OPC values • 96
- Show historical data chart • 128
- Start time (Change equipment status) • 39
- Start time (Get alarms) • 118
- Start time (Get OPC values) • 85
- Start time (Identify equipment) • 31
- Start time (Monitor numeric value) • 109
- Start time (Separate equipment) • 59
- Start time (Set OPC values) • 100
- Start time (Show historical data chart) • 129
- Start time (Trigger graph transition) • 49
- Start time (Unbind equipment) • 67
- Status change result (Change equipment status) • 38
- String value bundle (Get OPC values) • 83
- String value bundle (Set OPC values) • 99
- Trigger execution successful (Trigger graph transition) • 48
- Trigger graph transition • 47
- Unbind equipment • 66
- Unit of measure - Numeric value bundle (Get OPC values) • 83
- Unit of measure - Numeric value bundle (Set OPC values) • 98
- Value - Boolean value bundle (Get OPC values) • 82
- Value - Boolean value bundle (Set OPC values) • 97
- Value - Numeric value bundle (Get OPC values) • 83
- Value - Numeric value bundle (Set OPC values) • 98
- Value - String value bundle (Get OPC values) • 84
- Value - String value bundle (Set OPC values) • 99
- Value (Monitor numeric value) • 108

- Overall status (Get alarms, output) • 118
- Override recorded value (Get OPC values, parameter) • 74
- Override value definition (Set OPC values, parameter) • 91

P

- Parent entity of group (parameter)
 - Identify equipment • 24
 - Separate equipment • 55
- Process parameter
 - Alarm exception (Get alarms) • 116
 - Alarm property (Get alarms) • 114
 - Allowed change actions (Change equipment status) • 36
 - Allowed triggers (Trigger graph transition) • 45
 - Boolean value bundle (Get OPC values) • 74
 - Boolean value bundle (Set OPC values) • 93
 - Change equipment status • 35
 - Chart axis (Show historical data chart) • 126
 - Chart plot (Show historical data chart) • 124
 - Chart unavailable (Show historical data chart) • 128
 - Child entity to be separated (Separate equipment) • 56
 - Enter identifier manually (Identify equipment) • 28
 - Equipment data mismatch (Change equipment status) • 37
 - Equipment data mismatch (Trigger graph transition) • 46
 - Equipment data mismatch (Unbind equipment) • 65
 - Equipment status check (Identify equipment) • 27
 - Equipment unbind failed (Unbind equipment) • 65
 - Expected value configuration - Boolean value bundle (Get OPC values) • 75
 - Expected value configuration - String value bundle (Get OPC values) • 80
 - Expected value definition - Boolean value bundle (Get OPC values) • 76
 - Expected value definition - String value bundle (Get OPC values) • 81
 - Force status (Change equipment status) • 37
 - Get alarms • 114
 - Get OPC values • 72
 - Identification mode (Identify equipment) • 25

- Identified equipment entity (Change equipment status) • 35
 - Identified equipment entity (Get alarms) • 115
 - Identified equipment entity (Get OPC values) • 73
 - Identified equipment entity (Monitor numeric value) • 104
 - Identified equipment entity (Set OPC values) • 91
 - Identified equipment entity (Show historical data chart) • 124
 - Identified equipment entity (Trigger graph transition) • 45
 - Identified equipment entity (Unbind equipment) • 63
 - Identify equipment • 24
 - Identify in current binding context (Identify equipment) • 25
 - Input at equipment (Set OPC values) • 92
 - Instruction (Change equipment status) • 35
 - Instruction (Get alarms) • 114
 - Instruction (Get OPC values) • 73
 - Instruction (Identify equipment) • 24
 - Instruction (Monitor numeric value) • 104
 - Instruction (Separate equipment) • 35
 - Instruction (Set OPC values) • 90
 - Instruction (Show historical data chart) • 123
 - Instruction (Trigger graph transition) • 44
 - Instruction (Unbind equipment) • 35
 - Keep binding (Separate equipment) • 57
 - Keep sub-group (Separate equipment) • 56
 - L-H configuration - Numeric value bundle (Get OPC values) • 77
 - Limit definition - Numeric value bundle (Get OPC values) • 79
 - LL-HH configuration - Numeric value bundle (Get OPC values) • 78
 - Mode (Get alarms) • 115
 - Mode (Get OPC values) • 73
 - Mode (Identify equipment) • 24
 - Mode (Monitor numeric value) • 104
 - Mode (Separate equipment) • 64
 - Mode (Set OPC values) • 91
 - Mode (Trigger graph transition) • 44
 - Mode (Unbind equipment) • 64
 - Monitor numeric value • 104
 - Monitoring exception (Monitor numeric value) • 106
 - Numeric property (Monitor numeric value) • 105
 - Numeric value bundle (Get OPC values) • 76
 - Numeric value bundle (Set OPC values) • 94
 - Override recorded value (Get OPC values) • 74
 - Override value definition (Set OPC values) • 91
 - Parent entity of group (Identify equipment) • 24
 - Parent entity of group (Separate equipment) • 55
 - Property value check (Identify equipment) • 26
 - Separate Equipment • 55
 - Set OPC values • 90
 - Show children (Identify equipment) • 26
 - Show children (Separate equipment) • 56
 - Show children (Trigger graph transition) • 45
 - Show children (Unbind equipment) • 64
 - Show historical data chart • 123
 - Skip identification (Identify equipment) • 29
 - Skip separation (Separate equipment) • 57
 - Status transition failed (Trigger graph transition) • 47
 - Stop monitoring and record result (Monitor numeric value) • 107
 - String value bundle (Get OPC values) • 79
 - String value bundle (Set OPC values) • 95
 - Trigger graph transition • 44
 - Unbind equipment • 63
 - Undo identification (Identify equipment) • 28
 - Unforeseen resume (Get alarms) • 116
 - Unforeseen resume (Monitor numeric value) • 107
 - Property value check (Identify equipment, parameter) • 26
- ## R
- Result (Identify equipment, output) • 30
- ## S
- Selected trigger (Trigger graph transition, output) • 48
 - Separate equipment • 51
 - Output variable • 58
 - Process parameter • 55
 - Set OPC values • 87

Output variable • 96
Process parameter • 90
Show children (parameter)
Identify equipment • 26
Separate equipment • 56
Trigger graph transition • 45
Unbind equipment • 64
Show historical data chart • 121
Output variable • 128
Process parameter • 123
Skip identification (Identify equipment, parameter) • 29
Skip separation (Separate equipment, parameter) • 57
Start time (output)
Change equipment status • 39
Get alarms • 118
Get OPC values • 85
Identify equipment • 31
Monitor numeric value • 109
Separate equipment • 59
Set OPC values • 100
Show historical data chart • 129
Trigger graph transition • 49
Unbind equipment • 67
Status change result (Change equipment status, output) • 38
Status transition failed (Trigger graph transition, parameter) • 47
Stop monitoring and record result (Monitor numeric value, parameter) • 107
String value bundle (output)
Get OPC values • 83
Set OPC values • 99
String value bundle (parameter)
Get OPC values • 79
Set OPC values • 95

T

Trigger execution successful (Trigger graph transition, output) • 48
Trigger graph transition • 41
Output variable • 47

Process parameter • 44

U

Unbind equipment • 61
Output variable • 66
Process parameter • 63
Undo identification (Identify equipment, parameter) • 28
Unforeseen resume (parameter)
Get alarms • 116
Monitor numeric value • 107
Unit of measure - Numeric value bundle (output)
Get OPC values • 83
Set OPC values • 98

V

Value - Boolean value bundle (output)
Get OPC values • 82
Set OPC values • 97
Value - Numeric value bundle (output)
Get OPC values • 83
Set OPC values • 98
Value - String value bundle (output)
Get OPC values • 84
Set OPC values • 99
Value (Monitor numeric value, output) • 108