



DCS PHASES
RELEASE 10.01.00
USER MANUAL

PUBLICATION PSDC-UM003B-EN-E-MARCH-2021
Supersedes publication PSDC-UM003A-EN-E



Contact Rockwell See contact information provided in your maintenance contract.

Copyright Notice © 2021 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 10, 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	DCS Phases	1
	Typographical Conventions	2
	Structural Context.....	3
	Trigger Phases.....	4
	Materials Management with DCS Integration.....	13
	Synchronization Between PharmaSuite Operations and a DCS	13
Chapter 2	Create DCS Batch	15
	Execution	15
	Phase Design	18
	Process Parameters	19
	Output Variables.....	31
Chapter 3	Set DCS Parameters	37
	Execution	37
	Phase Design	40
	Process Parameters	41
	Output Variables.....	54
Chapter 4	Get DCS Parameters.....	59
	Execution	59
	Phase Design	62
	Process Parameters	63
	Output Variables.....	98
Chapter 5	Get DCS Alarms.....	107
	Execution	107
	Phase Design	109

	Process Parameters	110
	Output Variables.....	120
Chapter 6	DCS Alarm-based Trigger.....	123
	Execution	123
	Phase Design	123
	Process Parameters	124
	Output Variables.....	128
Chapter 7	Set Order Context	131
	Execution	131
	Phase Design	133
	Process Parameters	134
	Output Variables.....	141
Chapter 8	Show Consumed Material	145
	Execution	145
	Phase Design	147
	Process Parameters	147
	Output Variables.....	154
Chapter 9	Show Produced Material	157
	Execution	157
	Phase Design	159
	Process Parameters	159
	Output Variables.....	166
Chapter 10	Wait for Event (OES)	169
	Execution	169
	Phase Design	170
	Process Parameters	170
	Output Variables.....	171
Chapter 11	Wait for Event.....	175
	Execution	175

Phase Design	176
Process Parameters	176
Output Variables.....	183
Chapter 12 Send Event	187
Execution	187
Phase Design	189
Process Parameters	189
Output Variables.....	197
Index	199

Figure 1: Unit procedure with trigger phase and Event-triggered operation.....	5
Figure 2: Functionally invalid - Server-run operation with preceding operation	6
Figure 3: Functionally invalid - Event-triggered operation with preceding operation	7
Figure 4: Functionally valid - strictly parallel Server-run and Event-triggered operations	8
Figure 5: Functionally invalid - sequential Event-triggered operations with identical phase reference.....	9
Figure 6: Functionally valid - sequential Event-triggered operations with sequential phases to reference	10
Figure 7: Functionally invalid - looped Event-triggered operation after simultaneous branch.....	11
Figure 8: Functionally valid - loop around Server-run and Event-triggered operations	12
Figure 9: Create DCS batch during execution	16
Figure 10: User-triggered exceptions of Create DCS batch.....	17
Figure 11: Create DCS batch after phase completion	18
Figure 12: Create DCS batch in the Navigator	18
Figure 13: Set DCS parameters during execution	38
Figure 14: User-triggered exceptions of Set DCS parameters	39
Figure 15: Set DCS parameters after phase completion	40
Figure 16: Set DCS parameters in the Navigator	40
Figure 17: Get DCS parameters during execution.....	61
Figure 18: User-triggered exceptions of Get DCS parameters	61
Figure 19: Get DCS parameters after phase completion.....	62
Figure 20: Get DCS parameters in the Navigator.....	62
Figure 21: Get DCS alarms during execution.....	109
Figure 22: Get DCS alarms after phase completion.....	109
Figure 23: Get DCS alarms in the Navigator	109

Figure 24: List editor	112
Figure 25: Duration editor	113
Figure 26: List editor	125
Figure 27: Duration editor	126
Figure 28: Duration editor	126
Figure 29: Set order context during execution	132
Figure 30: User-triggered exceptions of Set order context.....	132
Figure 31: Set order context after phase completion	133
Figure 32: Set order context in the Navigator	133
Figure 33: Show consumed material during execution	146
Figure 34: User-triggered exceptions of Show consumed material	146
Figure 35: Show consumed material after phase completion	146
Figure 36: Show consumed material in the Navigator	146
Figure 37: Show produced material during execution.....	158
Figure 38: User-triggered exception of Show produced material	158
Figure 39: Show produced material after phase completion.....	158
Figure 40: Show produced material in the Navigator	158
Figure 41: Wait for event during execution.....	176
Figure 42: User-triggered exception of Wait for event	176
Figure 43: Wait for event after phase completion.....	176
Figure 44: Wait for event in the Navigator.....	176
Figure 45: Send event during execution (normal event).....	188
Figure 46: Send event during execution (abort event).....	188
Figure 47: Send event after phase completion (normal event).....	188
Figure 48: Send event after phase completion (abort event).....	188
Figure 49: Send event in the Navigator (normal event).....	188
Figure 50: Send event in the Navigator (abort event).....	188

DCS Phases

The DCS phases of PharmaSuite represent a collection of phases for communicating with a Distributed Control System (DCS). They provide functions for creating a batch on a DCS, for retrieving batch values and alarms from the DCS, and for configuring alarms to be retrieved at regular intervals.

The following phases are available:

- Create DCS Batch (page [15](#))
- Set DCS Parameters (page [37](#))
- Get DCS Parameters (page [59](#))
- Get DCS Alarms (page [107](#))
- DCS Alarm-based Trigger (page [123](#))
- Set Order Context (page [131](#))
- Show Consumed Material (page [145](#))
- Show Produced Material (page [157](#))
- Wait for Event (OES) (page [169](#))
- Wait for Event (page [175](#))
- Send Event (page [187](#))

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.

Structural Context

In addition to the configuration of the individual phases, there are also some configurations on different levels of the graph structure, which you need to observe to use the full potential of the phases.

Whether or not a recipe's or workflow's unit procedure is suitable for holding DCS trigger phases (page 4) does not only depend on its graph structure, but is also controlled by the capabilities assigned to the operations of the unit procedure:

- Capability prerequisites:
 - Only operations that hold both the **Event-triggered** capability and the **Trigger-enabled** capability can interpret the trigger events sent by a trigger phase.
 - Trigger phases become active and complete automatically and do not require user interaction. For this reason, they have no user interface and need to be located in an operation that runs on a server, invisible to operators who process orders with PharmaSuite for Production Execution. Thus, an operation that holds trigger phases needs to have the **Server-run** capability.
- Graph structure prerequisites:
 - A trigger phase can only send trigger events to an **Event-triggered** operation if the operation is active to process the triggers. Consequently, the **Event-triggered** operation must run at the same time as the phase from which it receives trigger events. This means that an operation holding trigger phases must be located after the same simultaneous branch on a parallel track (page 4) to the **Event-triggered** operation whose runs it controls.

In order to integrate MFC data of PharmaSuite with the material data provided by a DCS (page 13), suitable material input and output parameters need to be maintained with the DCS phases.

For integrated processing with PharmaSuite and a DCS or other operator-run operations, you can specify event messages to be exchanged between the systems (page 13) to ensure that they are synchronized at certain points of time in the process.

Trigger Phases

The placement of **DCS alarm-based trigger** phases in a recipe or workflow is determined by the capabilities their operations need to have and by the structural requirement to ensure their simultaneous activity with the **Event-triggered** operation to which they send trigger events.

TIP

Please note that

- an **Event-triggered** operation can reference several trigger phases for receiving trigger events from them.
- a trigger phase can be referenced by several **Event-triggered** operations for sending them trigger events.

The following rules apply with respect to the start and completion of trigger processing of each of the two phases:

- A trigger phase becomes active automatically as soon as processing reaches its operation, but trigger processing does not start until at least one **Event-triggered** operation that references the phase has become active as well. During execution this means that only when the template of an **Event-triggered** operation becomes visible in the Cockpit of PharmaSuite for Production Execution, does the trigger phase start processing and can send the trigger events, which create the runs of the operation.
- If no **Event-triggered** operation becomes active, the trigger phase completes automatically after its defined timeout period has elapsed.
- If trigger processing has started along with one of its **Event-triggered** operations, the phase continues to send trigger events until the last of its **Event-triggered** operations is completed, which happens when an operator removes the template of the **Event-triggered** operation from the Cockpit of PharmaSuite for Production Execution. Without a target to which it can send its trigger events, the trigger phase completes automatically.

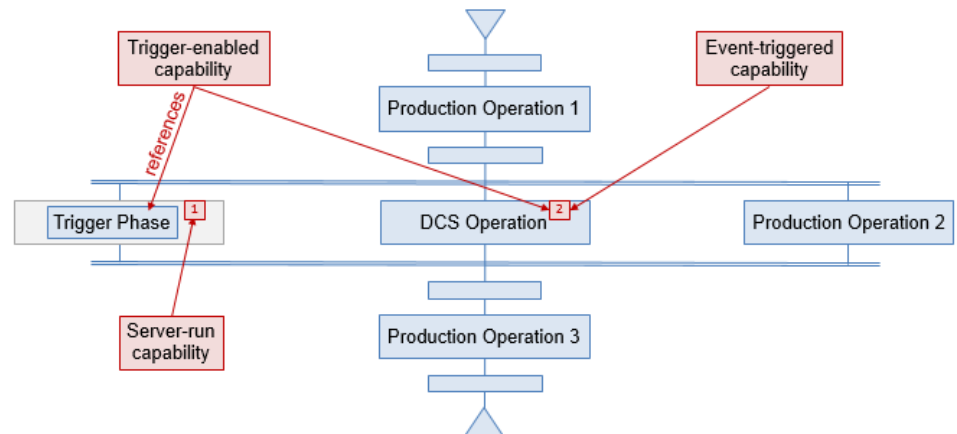


Figure 1: Unit procedure with trigger phase and Event-triggered operation

IMPORTANT

The general rules for building recipe structures as SFC graphs also govern unit procedures that hold trigger phases and **Event-triggered** operations. Thus, it is possible to build recipes that are valid from an SFC graph perspective, but do not meet the functional requirements made by processing use cases such as event-triggered DCS alarm retrieval. A recipe with functionally invalid structures is bound to cause serious issues during execution.

ISSUE: OPERATIONS NOT STRICTLY PARALLEL

Having operations precede either the **Server-run** operation with the trigger phases or the **Event-triggered** operation may lead to timing issues during execution.

- If the **Server-run** operation has a direct predecessor operation within the simultaneous branch, the system shows the following behavior during execution:
 - The template of the **Event-triggered** operation is visible in the Cockpit, but its trigger phase is not yet active to provide it with trigger events.
 - The **Server-run** operation holding the trigger phase can only become active when its preceding operation has been completed, thus causing an indeterminate delay of trigger processing.

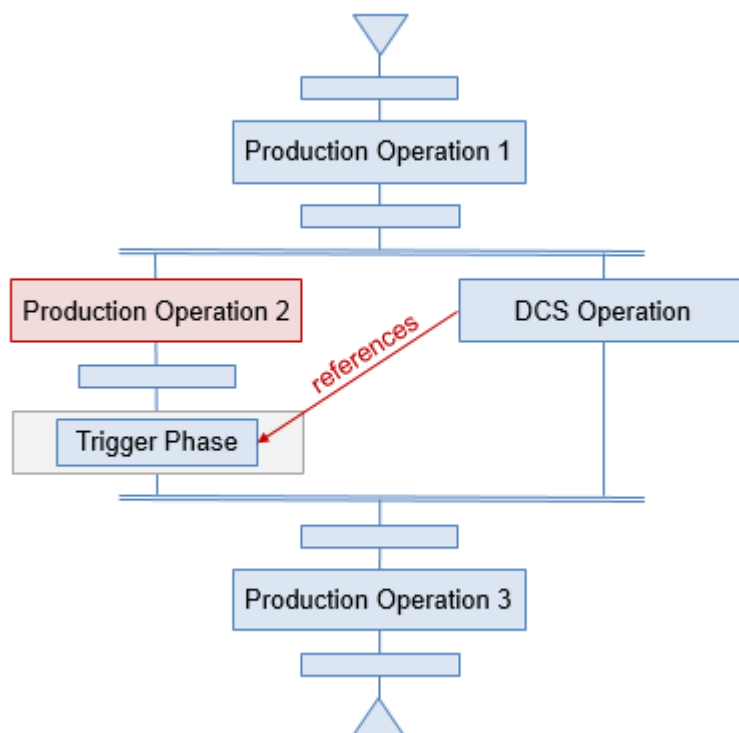


Figure 2: Functionally invalid - Server-run operation with preceding operation

- If the **Event-triggered** operation has a direct predecessor operation within the simultaneous branch, the system shows the following behavior during execution:
 - The trigger phase becomes active, but trigger processing does not start unless the operation preceding the **Event-triggered** operation has been completed and its template has become active.
 - The active trigger phase waits for its targeted **Event-triggered** operation to become active, but only until its defined timeout period has elapsed.
 - After the timeout period has elapsed, the trigger phase is completed automatically.
 - Thus, the indeterminate delay of the targeted **Event-triggered** operation can cause its entire trigger schedule to fail, since its referenced trigger phase has timed out before the **Event-triggered** operation has become active at all.

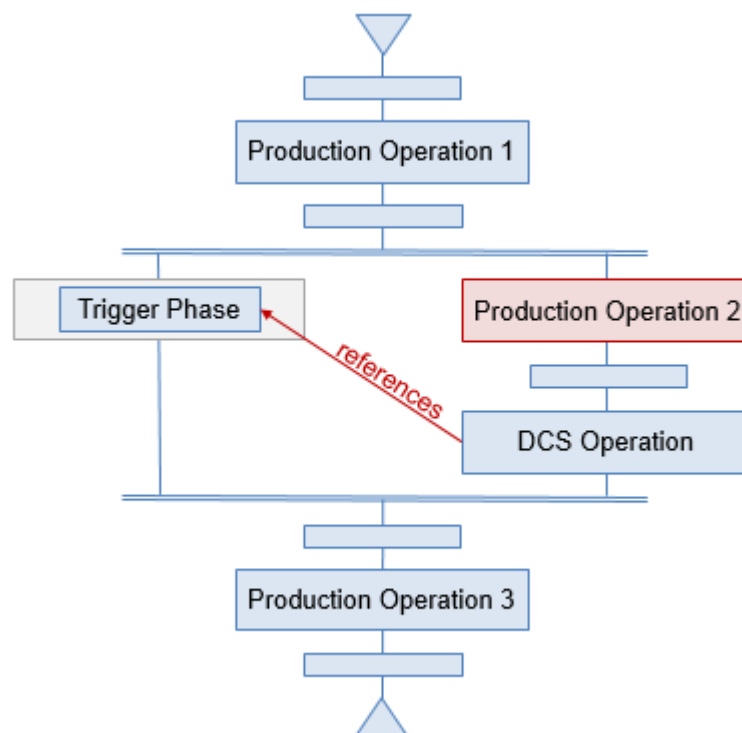


Figure 3: Functionally invalid - Event-triggered operation with preceding operation

TIP

To avoid issues of this type, place the predecessor operation before the simultaneous branch to establish a simultaneous activation of the **Server-run** operation holding the trigger phases and its targeted **Event-triggered** operation.

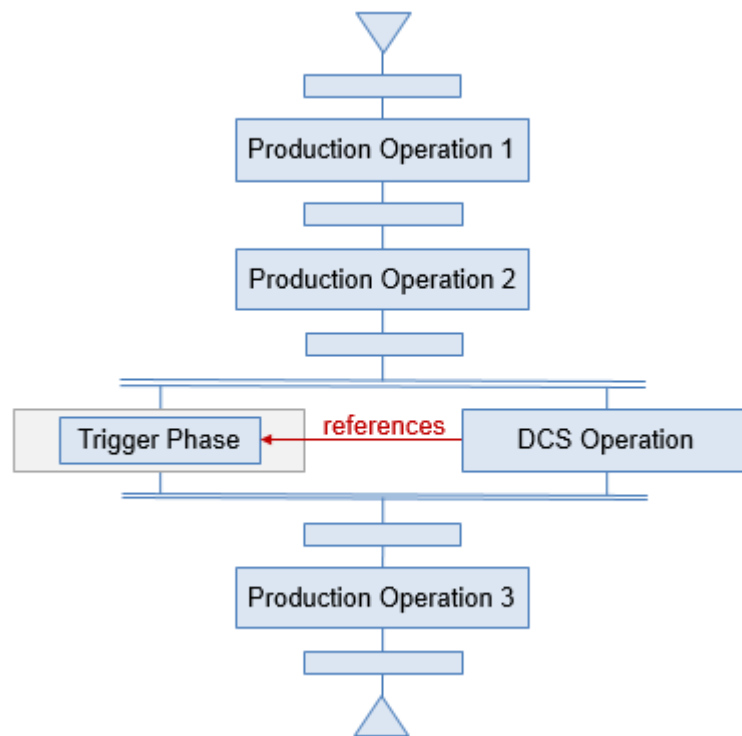


Figure 4: Functionally valid - strictly parallel Server-run and Event-triggered operations

ISSUE: SEQUENTIAL EVENT-TRIGGERED OPERATIONS WITH IDENTICAL PHASE REFERENCE

If your unit procedure has two **Event-triggered** operations that are located on the same track after the simultaneous branch and that both reference the same trigger phase, the system shows the following behavior during execution:

- The **Server-run** operation holding the trigger phase becomes active together with the first **Event-triggered** operation and the trigger phase begins to send trigger events as scheduled.
- Once the first **Event-triggered** operation has completed, the trigger phase is informed of this fact and completes automatically.
- As a consequence, when the second **Event-triggered** operation becomes active, its trigger phase is not available and its trigger schedule is bound to fail.

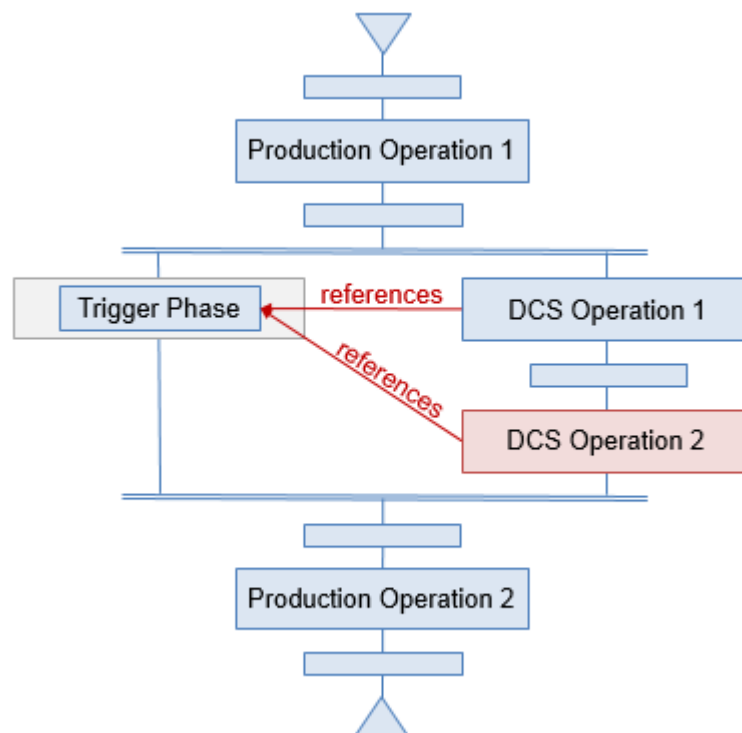


Figure 5: Functionally invalid - sequential Event-triggered operations with identical phase reference

TIP

To avoid issues of this type, place sequential trigger phases into the **Server-run** operation, one for each **Event-triggered** operation.

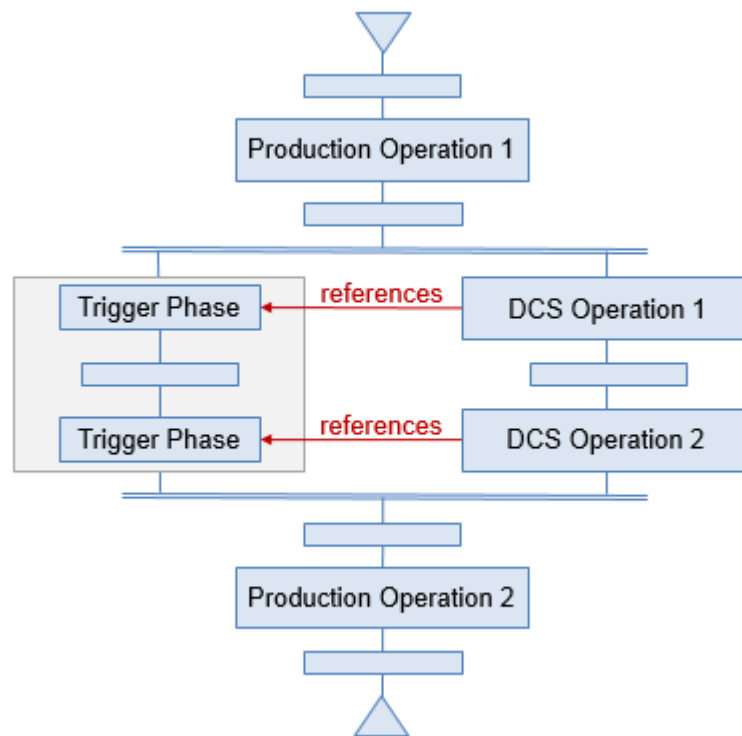


Figure 6: Functionally valid - sequential Event-triggered operations with sequential phases to reference

ISSUE: LOOPED EVENT-TRIGGERED OPERATION AFTER SIMULTANEOUS BRANCH

If your unit procedure has a loop only around its single **Event-triggered** operation, the system shows the following behavior during execution:

- The **Server-run** operation holding the trigger phase becomes active together with the **Event-triggered** operation and the trigger phase begins to send trigger events as scheduled.
- Once the **Event-triggered** operation has completed for the first time and reaches the loop's decision transitions, the trigger phase is informed of this fact and completes automatically.
- As a consequence, if the transition decides that the loop has to be passed through and the **Event-triggered** operation needs to be run again, its trigger phase is not available and its trigger schedule is bound to fail.

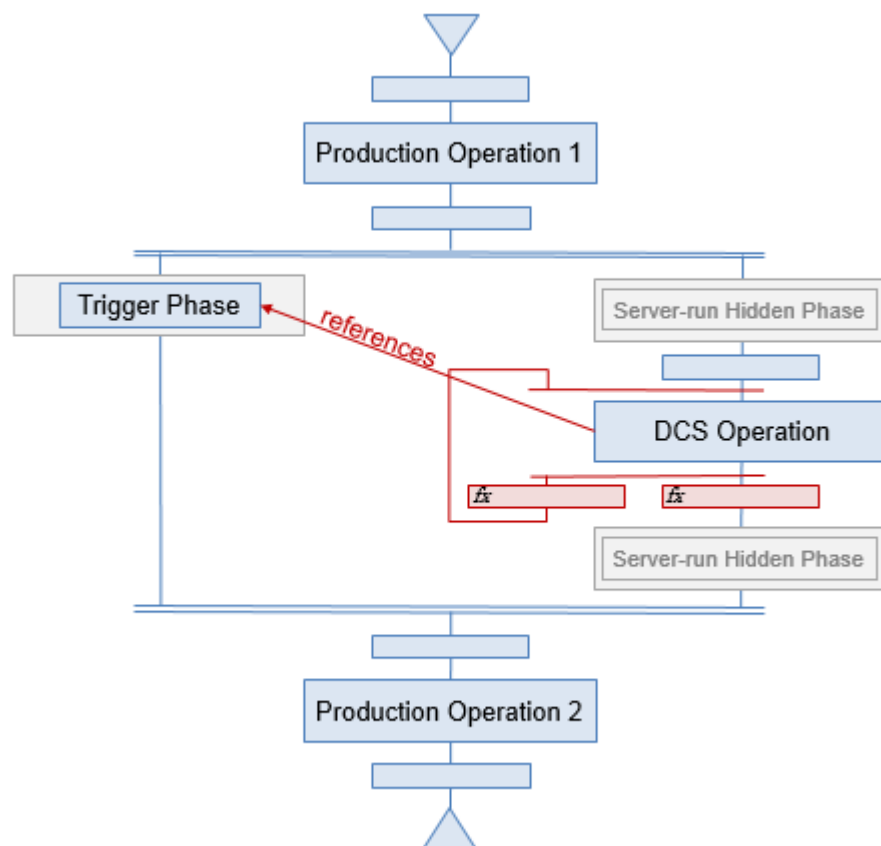


Figure 7: Functionally invalid - looped Event-triggered operation after simultaneous branch

TIP

To avoid issues of this type, enclose the entire simultaneous branch in the loop.

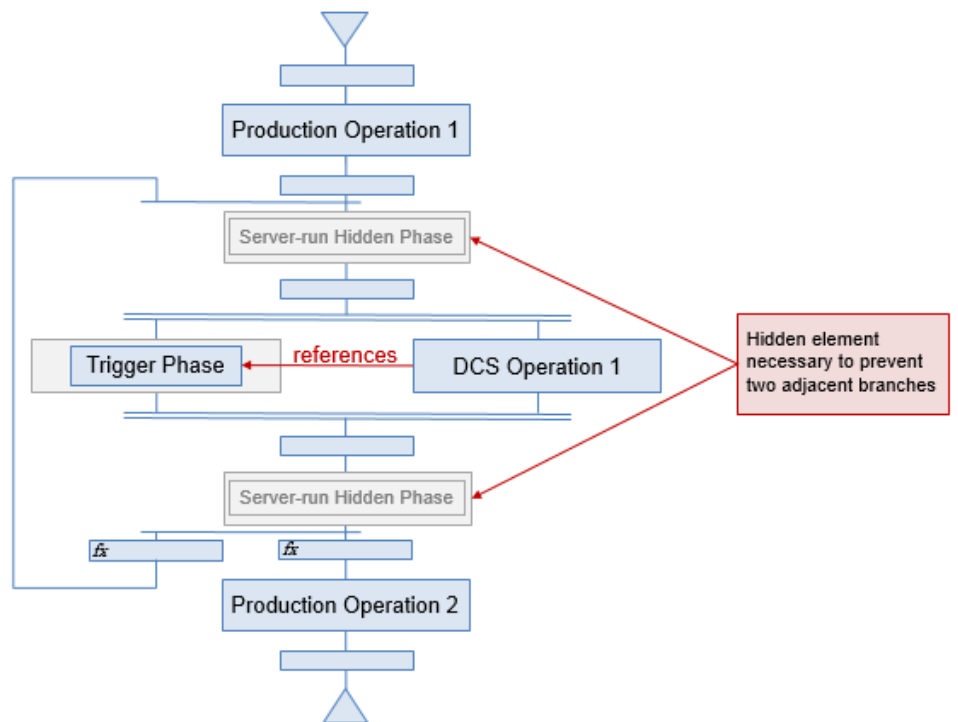


Figure 8: Functionally valid - loop around Server-run and Event-triggered operations

Materials Management with DCS Integration

In an automated and integrated environment, processing data needs to be exchanged between a DCS and PharmaSuite in order to document material balances that have occurred in the DCS in the batch report of the product.

The **Set order context** phase (page 131) downloads the context information of the current order to the DCS. Later on, material consumption and production data that is uploaded by the DCS is displayed by the **Show consumed material** phase (page 145) and the **Show produced material** phase (page 157). The operator can review the uploaded data and, if necessary, correct the data with user-triggered exceptions of the corresponding phases.

Synchronization Between PharmaSuite Operations and a DCS

In an automated and integrated environment, processing data needs to be exchanged between a DCS and PharmaSuite in order to synchronize the processes between the systems.

The **Wait for event (OES)** phase (page 169) receives an event message from the DCS, communicates with the **Send event** phase (page 187), and thus synchronizes its **server-run operation** with the DCS or an operator-run operation. The phase must be used in a unit procedure together with the **Send event** phase which is configured to send an abort event in order to avoid that the completion of the unit procedure is blocked by the **Wait for event (OES)** phase.

Contrary to the server-run phase, the **Wait for event** phase (page 175) is an operator-run phase and allows to synchronize its **operator-run operation** with the DCS or another operator-run operation.

The phases can also be configured to run asynchronously. In this case, a targeted **Wait for event** phase does not have to be active when the **Send event** phase or a DCS sends its request.

With the **Send event** phase, running **Wait for event (OES)** phases can be aborted.

Depending on the configuration, this applies to a specific **Wait for event (OES)** phase of the order or to all **Wait for event (OES)** phases of the common unit procedure.

Operator-run **Wait for event** phases provide a user-triggered exception for being aborted.

-
-
- FT PharmaSuite® BB - User Manual DCS Phases
-
-

Create DCS Batch

The **Create DCS batch** phase allows an operator to request the creation of a batch on a DCS.

It can be used for processing requirements, such as:

- Creating a batch to be executed on an automation system
Based on an existing master recipe and other product-specific parameters, the phase allows to create a DCS batch on a DCS.

Execution

In addition to the instruction text, the **Create DCS batch** phase displays the data required by the DCS system for creating a batch:

- the name of the DSC on which the batch is to be created,
- the definition data of the batch to be created:
 - Batch ID
 - Master recipe ID
 - Formula ID
 - Campaign ID
 - Scale to define the size of the batch
 - Description of the batch
- a table that indicates the unit binding and thus the unit on which the batch is to be processed,
- a table that lists all parameters and their values to be set on the DCS for batch processing.
- an information message that indicates the status of the batch creation, it includes the identifier of the created batch and its creation timestamp if the creation was successful.

For this reason the phase provides user-triggered exceptions as long as it is active.

- to create a batch manually when the first creation request has failed.
- to re-send the creation request and overwrite the previously created one. In this case, the system creates a new batch with a new identifier on the DCS and discards the existing batch.
- to override the data defined with
 - the batch definition, such as batch ID or master recipe ID,
 - the unit binding,
 - the parameter values.

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

- In the **Manual completion** mode, the operator manually triggers the creation of the batch.
- In the **Automatic completion** mode, the phase creates the batch and is completed automatically without any operator interaction.

After completion the phase displays the created batch with its data in the Execution Window.

The Navigator displays the identifier of the created batch.

Create the batch on the DCS.

DCS name:

JavaDCSMock

Batch ID:

BX57_V399

Master recipe ID:

ID_SRTD-100

Formula ID:

SR-100

Campaign ID:

2016-Q4


Scale:

100

Description:

Sonolin retard 100

Create



Unit class / step	Unit ID
Automated Tablet Press/Tableting Run	U207-22

Parameter	Value	UoM
Tablet Dimensions		9.0 mm
Tablet Form		Yes
Status		OK

Batch not created yet.

Confirm






Figure 9: Create DCS batch during execution

Re-send the DCS batch creation request. Confirm 

Create the DCS batch manually. Confirm 

Batch ID

Override a defined DCS parameter.

BatchID:

Current value BX57_V399

Override value

Master recipe ID:

Current value ID_SRTD-100

Override value

Formula ID:

Current value SR-100

Override value

Campaign ID:

Current value 2016-Q4

Override value


Scale:

Current value 100

Override value

Description:


Current value Sonolin retard 100

Override value Confirm 

Automated Tablet Press/Tableting Run

Override the defined unit binding:


Current unit ID U207-22

New unit ID Confirm 

Tablet Dimensions

Override the defined value:


Current value 9.0 mm

Override value mm Confirm 

Tablet Form

Override the defined value:

Current value Yes

Override value ☒ Yes ☐ No Confirm 

Status

Override the defined value:

Current value OK


Override value Confirm 

Figure 10: User-triggered exceptions of Create DCS batch

Figure 11: Create DCS batch after phase completion

Figure 12: Create DCS batch in the Navigator

Phase Design

The characteristics of the **Create DCS batch** 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 following four rows of the left and center columns contain the defined batch data.

In the right column of these rows, the phase provides the **Create** button.

The next rows display two tabular views that span all columns. The first one contains the data of the unit binding and the second one shows the defined parameters with their values.

The merged left and center columns of the bottom row display the information message on the status of the batch creation.

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

When the phase is completed it shows the same three-column, multi-row layout.

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 is automatically completed after a batch has been created successfully on the DCS.

DCS

Defines the name of the DCS system to be accessed for alarms retrieval.

Attribute	Type	Comment
Name	String	Logical name of the DCS to be used. The available entries correspond to the entries in the DCSNames list. Default setting: First entry in the list. Maximum length is 255 characters.

Definition

Defines the basic data of the batch to be created.

Attribute	Type	Comment
Batch ID	String	Defines the identifier of the batch to be created on the DCS. Maximum length is 250 characters. This attribute is required by the DCS Adapter.
Master recipe ID	String	Defines the identifier of the master recipe to be used. Maximum length is 200 characters. This attribute is required by the DCS Adapter.
Formula ID	String	Defines the identifier of the formula to be used. Maximum length is 200 characters.
Description	String	Defines the description of the batch to be created. Maximum length is 200 characters.
Campaign ID	String	Defines the identifier of the campaign to be used. Maximum length is 200 characters.
Scale	BigDecimal	Defines the scale of the batch in percent.

Create batch manually

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to create the DCS batch manually by typing the internal batch identifier of the DCS batch. If the identifier is unknown at the point, the phase uses DEFAULT_BATCH_ID as batch identifier to create the DCS batch on the DCS. If another batch with the defined batch data has been created before, it is discarded when the user confirms the exception and the new batch is created. It covers incidents when batch creation failed due to a connection issue to the DCS and needs to be executed again.

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.

Override DCS parameter

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to override the basic data given with the **Definition** parameter (page 20).

It covers incidents when the defined data is faulty and needs to be adjusted.

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.

Override bundle parameter

Represents a user-triggered exception that is accessible from the Exception Window.
The exception allows an operator to override the values defined with bundle parameters (page 23) of the **Boolean**, **Numeric**, or **String** data types.
It covers incidents when a defined value is faulty and needs to be adjusted.

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.

Override unit binding

Represents a user-triggered exception that is accessible from the Exception Window.
The exception allows an operator to override data defined with a bundle parameter (page 23) of the **Unit binding** type.
It covers incidents when batch execution has to be performed on a different unit than originally intended and the value given with the **Unit ID** attribute needs to be adapted for this reason.

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.

Re-send creation request

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to send additional creation requests to the DCS after the first creation request has been processed and the **Create** button is no longer enabled. If another batch with the defined batch data has been created before, it is discarded when the user confirms the exception and the new batch is created. It covers incidents when a batch needs to be re-created with adjusted data, such as a different unit binding.

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.

Parameter bundles

In addition to the permanent process parameters that are always present, the **Create DCS batch** phase provides parameter bundles as optional process parameters, which you can insert if required.

You can add process parameter bundles for up to 50 values of four different data types (**Boolean**, **Numeric**, **String**, **Unit binding**) to the **Create DCS batch** phase.



ADDING PARAMETER BUNDLES

1. Click the **Add parameter** 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 the 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 parameter with its value of the **Boolean** data type that is to be set on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be used. Maximum length is 200 characters.
Value	Boolean	Defines the value of the parameter.

Numeric value

Indicates the parameter with its value of the **Numeric** data type that is to be set on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be used. Maximum length is 200 characters.
Value	BigDecimal	Defines the value of the parameter.
UoM	Unit of measure	Must match a unit of measure available within PharmaSuite.

String Value

Indicates the parameter with its value of the **String** data type that is to be set on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be used. Maximum length is 200 characters.
Value	String	Defines the value of the parameter. Maximum length is 200 characters.

Unit binding

Indicates the unit binding data that is to be set on the DCS.

Attribute	Type	Comment
Unit class/step	String	Defines the required unit class or step. Maximum length is 200 characters.
Unit ID	String	Defines the identifier of the unit to be used. Maximum length is 200 characters.

Instruction links

In addition to the permanent process parameters that are always present, the **Create DCS batch** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.







ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.
The system adds the instruction link parameter and the first link definition row to the list of parameters.
Instruction link parameters are generally inserted below all other parameters.
Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
 - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
 - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa. The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.
5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:
 -  adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.
 -  deletes the currently selected rows.
 -  moves the currently selected row one row up.
 -  moves the currently selected row one row down.



REMOVING INSTRUCTION LINKS

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction link parameter.

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

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	<p>Instruction text to be displayed.</p> <p>For any text enclosed in curly brackets you can define a hyperlink with the Instruction link definition process parameter (page 27).</p> <p>Example: Refer to {SOP1270} for guidance.</p> <p>Maximum length is 2000 characters (including HTML tags).</p>

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 27). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

`https://rockwellautomation.com`

`file:///c:/SOP/SOP1492.pdf`

Attribute	Type	Comment
Link text	Text	<p>Text to be used as link.</p> <p>For any text enclosed in curly brackets within the instruction text you can define a link with the Link URL attribute.</p> <p>Including the brackets in the link text is optional.</p> <p>Maximum length is 80 characters.</p>
Link URL	Text	<p>URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system.</p> <p>Maximum length is 256 characters.</p>

Instruction tables

In addition to the permanent process parameters that are always present, the **Create DCS batch** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.







ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
The system adds the parameter definition and the first table row to the list of parameters.
Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction table as last parameter.
 - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
 - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
 - Select the number of columns to define the layout.
 - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
 - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:
 -  adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.
 -  deletes the currently selected rows.
 -  moves the currently selected row one row up.
 -  moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

**REMOVING INSTRUCTION TABLES**

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction table with its definition and rows.

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

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: 1 column, 2 columns, 3 columns, 4 columns, 5 columns . Default setting: 1 column .
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. Restriction: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Create DCS batch** phase provides the following output variables:

Batch 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 DCS batch that was created on the DCS.

Campaign 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 campaign that was used to create the DCS batch.

Internal batch 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 internal batch that depends on the used DCS.

Master recipe 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 master recipe that was used to create the DCS batch.

Scale

- Data type: BigDecimal, floating point number that allows calculating with greater precision than Float.
- Usage: The output variable provides the scale that was used to create the DCS batch.

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 - Parameter

- 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 parameter that was used to create the DCS batch.

Boolean value - Value

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable provides the value of the parameter that was used to create the DCS batch.

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 - Parameter

- 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 parameter that was used to create the DCS batch.

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 parameter that was used to create the DCS batch.

Numeric Value - Value

- Data type: BigDecimal, floating point number that allows calculating with greater precision than Float.
- Usage: The output variable provides the value of the parameter that was used to create the DCS batch.

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 - Parameter

- 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 parameter that was used to create the DCS batch.

String value - Value

- 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 value of the parameter that was used to create the DCS batch.

Unit binding

If you have added unit binding 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 unit binding bundles:

Unit binding - Unit class

- 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 unit class that was used to create the DCS batch.

Unit binding - Unit 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 unit that was used to create the DCS batch.

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"**.

Set DCS Parameters

The **Set DCS parameters** phase allows an operator to set parameter values on a DCS.

It can be used for processing requirements, such as:

- GxP-relevant parameters for a process are maintained with the MES master recipe and needs to be transferred to the automation system (e.g. speed and duration for a blender).

Execution

In addition to the instruction text, the **Set DCS parameters** phase displays the following data:

- the name of the DCS on which the parameters are to be set
- the basic data of the batch for which the parameters are to be set
 - Batch ID
 - Unit ID
 - Product ID
 - Recipe ID
 - Data target
 - Module IDs
- an information message that shows the status of the set operation
 - **Parameters not set yet** indicates that no set operation has been performed yet or that the parameter definition or the parameter data was overridden after a set operation,
 - **Error when setting parameters** indicates that the set operation was not successful.
 - **Error when re-setting parameters** indicates that a re-set operation initiated by a user-triggered exception was not successful.
 - **Sent at: <timestamp>** indicates that a set operation has been performed and when it took place.
- a table that lists all parameters and their values.

The **Set** button is disabled after the set operation has been performed regardless of the operation's success. Any further requests or changes to the defined data are considered exceptions.

- to re-send the request to set the DCS parameter values on the DCS. With the exception, the phase overwrites any previously set parameter values,
- to confirm that the parameter values have been set manually on the DCS,
- to override the data set with the **Definition** process parameter (page 42) and any of the process parameter bundles (page 45).

The Navigator displays the identifier of the updated batch.

Set the data on the DCS.

DCS name:

JavaDCSMock

Batch ID:

BX57_V399

Unit ID:

U207-22

Product ID:

D130-01

Recipe ID:

ID_SRTD-100

Data target:

Target 2

Module IDs:

Parameters not set yet.

Parameter	Value
Tablet Dimensions	9.0
Tablet Form	Yes
Status	OK
Order Start	11/26/2018 6:13:30 PM CET
Order Duration	7h

Set

Send parameters to the DCS.
Confirm

Set the DCS parameters manually.
Confirm

Override a definition parameter.

Batch ID:

Current value
BX57_V399

Override value

Unit ID:

Current value
U207-22

Override value

Recipe ID:

Current value
ID_SRTD-100

Override value

Product ID:

Current value
D130-01

Override value

Product ID:

Current value
D130-01

Override value

Data target:

Current value
Target 2

Override value

Module IDs:

Current value
N/A

Override value

Confirm

Tablet Dimensions

Override the defined value:

Current value
9.0

Override value

Confirm

Tablet Form

Override the defined value:

Current value
Yes

Override value

Yes

No

Confirm

Status

Override the defined value:

Current value
OK

Override value

Confirm

Order Start

Override the defined value:

Current value
11/08/2018 06:13:30 PM CET

Override value

Confirm

Order Duration

Override the defined value:

Current value
7h

Override value

Confirm

Figure 14: User-triggered exceptions of Set DCS parameters

Set the data on the DCS.

DCS name:
JavaDCSMock
Batch ID:
BX57_V399

Unit ID:
U207-22
Product ID:
D130-01

Recipe ID:
ID_SRTD-100
Data target:
Target 2

Module IDs:

Set

Sent at: 11/26/2018 01:46:23 PM CET

Parameter	Value
Tablet Dimensions	9.0
Tablet Form	Yes
Status	OK
Order Start	11/26/2018 6:13:30 PM CET
Order Duration	7h

Confirm

Figure 15: Set DCS parameters after phase completion

Integrated DCS Run [1.1]
Set DCS Data

BX57_V399

Figure 16: Set DCS parameters in the Navigator

Phase Design

The characteristics of the **Set DCS parameters** 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 following four rows of the left and center columns, the phase displays the basic data of the batch. The right column contains the **Set** button. The merged columns of the next row contain the information message indicating the operation's status. The next rows display a tabular view of relevant data of each parameter to be set and its values, spanning all columns. The right column of the bottom row contains the **Confirm** button.

When the phase is completed it shows the same three-column, multi-row layout. 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 is automatically completed after the parameter values have been set successfully on the DCS.

DCS

Defines the name of the DCS system to be accessed for value setting.

Attribute	Type	Comment
Name	String	Logical name of the DCS to be used. The available entries correspond to the entries in the DCSNames list. Default setting: First entry in the list. Maximum length is 255 characters.

Definition

Defines the basic batch data to be set on the DCS.

Attribute	Type	Comment
Batch ID	String	Optional parameter to define the identifier of the batch whose parameter values will be set on the DCS. Maximum length is 250 characters.
Unit ID	String	Optional parameter to define the identifier of the unit to be used. Maximum length is 250 characters.
Module IDs	Text (structured)	Optional parameter to define the list of equipment module IDs and control module IDs. Maximum length is 2000 characters.
Product ID	String	Optional parameter to define the identifier of the product. Maximum length is 250 characters.
Recipe ID	String	Optional parameter to define the identifier of the recipe. Maximum length is 250 characters.
Data target	Choice list	Optional parameter to define how the message is processed in the Manufacturing Service Bus (MSB) (e.g. SMAC, ULDL, RMS). Maximum length is 250 characters.

Re-send request

Represents a user-triggered exception that is accessible from the Exception Window.

The exception allows an operator to send additional requests to the DCS after the first request has been processed and the **Set** button is no longer enabled.

If the defined batch data has already been set to different values before, they are overwritten when the user confirms the exception.

It covers incidents when batch data or parameter values need to be updated with adjusted data, such as a different unit ID or a different parameter value.

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.

Set DCS parameters manually

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to confirm that the required data has been set manually at the DCS.

It covers incidents when setting the parameters has failed and cannot be repeated.

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.

Override bundle parameter

Represents a user-triggered exception that is accessible from the Exception Window.
 The exception allows an operator to override a value set on the DCS.
 It covers incidents when a set value is faulty and needs to be adjusted.

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.

Override definition

Represents a user-triggered exception that is accessible from the Exception Window.
 The exception allows an operator to override a batch definition set on the DCS.
 It covers incidents when a defined value is faulty and needs to be adjusted.

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.

Parameter bundles

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

You can add process parameter bundles for up to 50 values of five different data types (**Boolean**, **Duration**, **Numeric**, **String**, **Timestamp**) to the **Set DCS parameters** phase.



ADDING PARAMETER BUNDLES

1. Click the **Add parameter** 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 the 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 - Master (bundle identifier)

Indicates the parameter with its value of the **Boolean** data type that is to be set on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read. Maximum length is 200 characters.
Value	Boolean	Defines the value of the parameter.

Boolean value - Category

Represents an optional parameter classification.

Attribute	Type	Comment
Category	Choice list	Defines the category of the bundle. Available settings: ---, P1, P2, P3, P4.

Duration value - Master (bundle identifier)

Indicates the parameter with its value of the **Duration** data type that is to be set on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be used. Maximum length is 200 characters.
Value	Duration	Defines the value of the parameter.

Duration value - Category

Represents an optional parameter classification.

Attribute	Type	Comment
Category	Choice list	Defines the category of the bundle. Available settings: ---, P1, P2, P3, P4.

Numeric value - Master (bundle identifier)

Indicates the parameter with its value of the **Numeric** data type that is to be set on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be used. Maximum length is 200 characters.
Value	BigDecimal	Defines the value of the parameter.

Numeric value - Category

Represents an optional parameter classification.

Attribute	Type	Comment
Category	Choice list	Defines the category of the bundle. Available settings: ---, P1, P2, P3, P4.

String Value - Master (bundle identifier)

Indicates the parameter with its value of the **String** data type that is to be set on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be used. Maximum length is 200 characters.
Value	String	Defines the value of the parameter. Maximum length is 2000 characters.

String value - Category

Represents an optional parameter classification.

Attribute	Type	Comment
Category	Choice list	Defines the category of the bundle. Available settings: ---, P1, P2, P3, P4.

Timestamp value - Master (bundle identifier)

Indicates the parameter with its value of the **Timestamp** data type that is to be set on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be used. Maximum length is 200 characters.
Value	Timestamp	Defines the value of the parameter.

Timestamp value - Category

Represents an optional parameter classification.

Attribute	Type	Comment
Category	Choice list	Defines the category of the bundle. Available settings: ---, P1, P2, P3, P4.

Instruction links

In addition to the permanent process parameters that are always present, the **Set DCS parameters** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
The system adds the instruction link parameter and the first link definition row to the list of parameters.
Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
 - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
 - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa. The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.
5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:



adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.



deletes the currently selected rows.



moves the currently selected row one row up.



moves the currently selected row one row down.

**REMOVING INSTRUCTION LINKS**

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction link parameter.

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

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	<p>Instruction text to be displayed.</p> <p>For any text enclosed in curly brackets you can define a hyperlink with the Instruction link definition process parameter (page 50).</p> <p>Example: Refer to {SOP1270} for guidance.</p> <p>Maximum length is 2000 characters (including HTML tags).</p>

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 50). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

`https://rockwellautomation.com`

`file:///c:/SOP/SOP1492.pdf`

Attribute	Type	Comment
Link text	Text	<p>Text to be used as link.</p> <p>For any text enclosed in curly brackets within the instruction text you can define a link with the Link URL attribute.</p> <p>Including the brackets in the link text is optional.</p> <p>Maximum length is 80 characters.</p>
Link URL	Text	<p>URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system.</p> <p>Maximum length is 256 characters.</p>

Instruction tables

In addition to the permanent process parameters that are always present, the **Set DCS parameters** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.







ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
The system adds the parameter definition and the first table row to the list of parameters.
Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction table as last parameter.
 - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
 - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
 - Select the number of columns to define the layout.
 - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
 - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:
 -  adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.
 -  deletes the currently selected rows.
 -  moves the currently selected row one row up.
 -  moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.



REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction table with its definition and rows.

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

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: 1 column , 2 columns , 3 columns , 4 columns , 5 columns . Default setting: 1 column .
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. Restriction: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Set DCS parameters** phase provides the following output variables:

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 - Parameter

- 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 parameter that was used to set the DCS parameter values.

Boolean value - Value

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable provides the value of the parameter that was set on the DCS.

Duration value

If you have added Duration 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 Duration value bundles:

Duration value - Parameter

- 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 parameter that was used to set the DCS parameter values.

Duration value - Value

- Data type: Duration, used for displaying time spans and for time-related calculations.
- Usage: The output variable provides the value of the parameter that was set on the DCS.

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 - Parameter

- 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 parameter that was used to set the DCS parameter values.

Numeric value - Value

- Data type: BigDecimal, floating point number that allows calculating with greater precision than Float.
- Usage: The output variable provides the value of the parameter that was set on the DCS.

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 - Parameter

- 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 parameter that was used to set the DCS parameter values.

String value - Value

- 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 value of the parameter that was set on the DCS.

Timestamp value

If you have added Timestamp 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 Timestamp value bundles:

Timestamp value - Parameter

- 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 parameter that was used to set the DCS parameter values.

Timestamp value - Value

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
- Usage: The output variable provides the value of the parameter that was set on the DCS.

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 DCS Parameters

The **Get DCS parameters** phase allows an operator to retrieve automation system parameter values through message queues.

It can be used for processing requirements, such as:

- Document batch relevant parameters in the batch report that were used by the automated system and not set by the MES.
- Evaluate, document, and generate exceptions for batch-relevant parameters that are not within the limits defined by further automation parameters.

Execution

In addition to the instruction text, the **Get DCS parameters** phase displays the following data:

- the filter criteria applied when the system retrieves data from the DCS
 - DCS name
 - Batch ID
 - Unit ID
 - Product ID
 - Recipe ID
 - Data source
 - Module IDs
- an information message that shows the status of the retrieval
 - **Get action not performed yet** indicates that no retrieval has been performed yet,
 - **Timestamp of last Get action: <timestamp>** indicates that a retrieval has been performed and when the last retrieval took place.
- a table that lists all parameters and their values and limits, if defined.

When the operator taps the **Get** button the phase accesses the DCS and retrieves the specified data. For Numeric, Timestamp, and Duration 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.

The **Get** button remains enabled until all values have been retrieved once. Values that have already been retrieved successfully are not retrieved again when the operator re-taps the **Get** button. Manually overridden values are also excluded from being retrieved. After all values have been retrieved once or have been overridden manually, the **Get** button is disabled. Any further changes to the retrieved values are considered exceptions.

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

TIP

Please note that an additional table with up to five columns and 50 rows may be shown below the parameter table. Its data supplied by the DCS. The layout and content of the additional table is solely controlled by the DCS and cannot be configured during phase configuration.

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

- In the **Manual completion** mode, the operator manually triggers retrieving the parameters.
- In the **Automatic completion** mode, the phase retrieves the parameters 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.

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

The Navigator displays the identifier of the batch whose values were retrieved.


Collect the data from the DCS.

DCS name: JavaDCSMock Batch ID: BX57_V399
Unit ID: U207-22 Product ID: D130-01
Recipe ID: ID_SRTD-100 Data source: Source 2

Module IDs:

Timestamp of last Get action: 11/26/2018 05:22:11 PM CET

Parameter	Parameter description	Expected	Limits (LL L)	Value	Limits (H HH)
Tablet Dimensions	Tablet dimensions value		8.8 8.9	9.0	9.1 9.2
Tablet Form	Tablet form (round)	Yes		Yes	
Status	Process status	OK		OK	
Order Start	Date and time of order start		11/24/2018 12:00:00 AM CET 11/25/2018 12:00:00 AM CET	11/26/2018 05:22:11 PM CET	11/29/2018 12:00:00 AM CET 11/30/2018 12:00:00 AM CET
Order Duration	Run duration of the order		3h 4h	7h	11h 12h

Get 


Confirm 


Figure 17: Get DCS parameters during execution

Tablet Dimensions (Tablet dimensions value)

Override recorded value:

Current value: 9.0

New value:


Confirm 

Tablet Form (Tablet form (round))

Override recorded value:

Current value: Yes

New value: ☒ Yes ☐ No


Confirm 

Status (Process status)

Override recorded value:

Current value: OK

New value:


Confirm 

Order Start (Date and time of order start)

Override recorded value:

Current value: 11/26/2018 05:22:11 PM CET

New value:

Confirm 

Order Duration (Run duration of the order)

Override recorded value:

Current value: 7h

New value:


Confirm 

Figure 18: User-triggered exceptions of Get DCS parameters

Collect the data from the DCS.

DCS name: JavaDCSMock Batch ID: BX57_V399
Unit ID: U207-22 Product ID: D130-01
Recipe ID: ID_SRTD-100 Data source: Source 2
Module IDs:

Timestamp of last Get action: 11/26/2018 05:22:11 PM CET

Parameter	Parameter description	Expected	Limits (LL L)	Value	Limits (H HH)
Tablet Dimensions	Tablet dimensions value		8.8 8.9	9.0	9.1 9.2
Tablet Form	Tablet form (round)	Yes		Yes	
Status	Process status	OK		OK	
Order Start	Date and time of order start		11/24/2018 12:00:00 AM CET 11/25/2018 12:00:00 AM CET	11/26/2018 05:22:11 PM CET	11/29/2018 12:00:00 AM CET 11/30/2018 12:00:00 AM CET
Order Duration	Run duration of the order		3h 4h	7h	11h 12h

Get

Confirm

Figure 19: Get DCS parameters after phase completion

Integrated DCS Run [1.1]
Collect DCS Data

BX57_V399

Figure 20: Get DCS parameters in the Navigator

Phase Design

The characteristics of the **Get DCS parameters** 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 following five rows of the left and center columns, the phase displays the filter criteria configured for data retrieval. The right column contains the **Get** button. The merged columns of the next row contain the information message indicating the retrieval status.

The next rows display a tabular view of relevant data of each retrieved parameter 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. When an operator adds an exception, the phase displays an exception marker at the affected value. The right column of the bottom row contains the **Confirm** button.

When the phase is completed it shows the same three-column, multi-row layout.

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).

DCS

Defines the name of the DCS system to be accessed for value retrieval.

Attribute	Type	Comment
Name	String	Logical name of the DCS to be used. The available entries correspond to the entries in the DCSNames list. Default setting: First entry in the list. Maximum length is 255 characters.

Definition

Defines the filter criteria by which values will be retrieved.

Attribute	Type	Comment
Batch ID	String	Optional parameter to define the identifier of the batch to be used for data retrieval. Maximum length is 250 characters.
Unit ID	String	Optional parameter to define the identifier of the unit to be used. Maximum length is 250 characters.
Module IDs	Text (structured)	Optional parameter to define the list of equipment module IDs and control module IDs. Maximum length is 2000 characters.

Attribute	Type	Comment
Product ID	String	Optional parameter to define the identifier of the product. Maximum length is 250 characters.
Recipe ID	String	Optional parameter to define the identifier of the recipe. Maximum length is 250 characters.
Data source	Choice list	Optional parameter to define how the message is processed in the Manufacturing Service Bus (MSB) (e.g. SMAC, ULDL, RMS). Maximum length is 250 characters.

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 is automatically completed after the batch values have been retrieved successfully from the DCS batch.

Override DCS parameter value

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to override a value retrieved from the DCS. It covers incidents when a retrieved value is faulty and needs to be adjusted or when the operator needs to confirm that a description, expected value, or limits that should have been retrieved from the DCS are missing.

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.

Parameter bundles

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

You can add process parameter bundles for up to 50 values of ten different data types (**Boolean**, **Boolean (Extended)**, **Duration**, **Duration (Extended)**, **Numeric**, **Numeric (Extended)**, **String**, **String (Extended)**, **Timestamp**, **Timestamp (Extended)**) to the **Get DCS parameters** phase.



ADDING PARAMETER BUNDLES

1. Click the **Add parameter** 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 the 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 - Master (bundle identifier)

Specifies the parameter identifier and description of the **Boolean** data type that is to be retrieved from the DCS. The **Parameter path** indicates where the value is located on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read. Maximum length is 200 characters.
Parameter description	String	Defines an alias for the parameter. Maximum length is 200 characters.
Parameter path	String	Defines the path of the parameter. Maximum length is 400 characters.

Boolean value - Expected value configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS must be checked against an expected value.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the Value attribute of the Expected value definition process parameter (page 67) is set. If it is not set, the validation will fail. 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.

Boolean value - 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 .

Boolean value - Expected value visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows an empty cell.

Attribute	Type	Comment
Enabled	Flag	Controls if the expected value is visible during execution. Default setting: Yes

Boolean value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt, Report, Recipe, Status change) correspond to the entries in the DCSParаметerType list. Maximum length is 200 characters.

Boolean (extended) value - Master (bundle identifier)

In an extended data type, all data is retrieved from the DCS. Thus, the **Parameter** and **Parameter for description** attributes indicate the names the data has on the DCS. The **Common path** attribute indicates where on the DCS the two named parameters and the parameter for the expected value are located.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read on the DCS. Maximum length is 200 characters.
Parameter for description	String	Defines the name of the description parameter on the DCS. Maximum length is 200 characters.
Common path	String	Defines the path of all parameters of the bundle on the DCS. Maximum length is 400 characters.

Boolean (extended) value - Expected value configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS must be checked against an expected value.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the Value attribute of the Expected value definition process parameter (page 69) is set. If it is not set, the validation will fail. 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.

Boolean (extended) value - 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 name of the expected value parameter on the DCS. Maximum length is 200 characters.

Boolean (extended) value - Expected value visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows an empty cell.

Attribute	Type	Comment
Enabled	Flag	Controls if the expected value is visible during execution. Default setting: Yes

Boolean (extended) value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt, Report, Recipe, Status change) correspond to the entries in the DCSParаметerType list. Maximum length is 200 characters.

Duration value - Master (bundle identifier)

Specifies the parameter identifier and description of the **Duration** data type that is to be retrieved from the DCS. The **Parameter path** indicates where the value is located on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read. Maximum length is 200 characters.
Parameter description	String	Defines an alias for the parameter. Maximum length is 200 characters.
Parameter path	String	Defines the path of the parameter. Maximum length is 400 characters.

Duration value - L-H configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 72). If both checks are enabled, they are performed in the following order:

1. LL-HH (defined with the **LL-HH configuration** process parameter (page 71))
2. L-H

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the L limit or H limit attributes of the Limit definition process parameter (page 72) are set. If they are not set, the validation will fail. 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.

Duration value - LL-HH configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 72). If both checks are enabled, they are performed in the following order:

1. LL-HH
2. L-H (defined with the **L-H configuration** process parameter (page 70))

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the LL limit or HH limit attributes of the Limit definition process parameter (page 72) are set. If they are not set, the validation will fail. 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.

Duration value - Limit definition

Limits are defined as absolute values. Make sure that the limits are strictly sequential and do not overlap, so that

$$\blacksquare \quad \text{LL limit} < \text{L limit} < \text{H limit} < \text{HH limit}$$

Attribute	Type	Comment
LL limit	Duration	Define the values of the lower limits (including the values themselves).
L limit	Duration	
H limit	Duration	Define the values of the upper limits (including the values themselves).
HH limit	Duration	

Duration value - Limits visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows empty cells.

Attribute	Type	Comment
Enabled	Flag	Controls if the limits are visible during execution. Default setting: Yes

Duration value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt, Report, Recipe, Status change) correspond to the entries in the DCSParameterType list. Maximum length is 200 characters.

Duration (extended) value - Master (bundle identifier)

In an extended data type, all data is retrieved from the DCS. Thus, the **Parameter** and **Parameter for description** attributes indicate the names the data has on the DCS. The **Common path** attribute indicates where on the DCS the two named parameters and the parameters for the limits are located.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read on the DCS. Maximum length is 200 characters.
Parameter for description	String	Defines the name of the description parameter on the DCS. Maximum length is 200 characters.
Common path	String	Defines the path of all parameters of the bundle on the DCS. Maximum length is 400 characters.

Duration (extended) value - L-H configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 75). If both checks are enabled, they are performed in the following order:

1. LL-HH (defined with the **LL-HH configuration** process parameter (page 74))
2. L-H

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the L limit or H limit attributes of the Limit definition process parameter (page 75) are set. If they are not set, the validation will fail. 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.

Duration (extended) value - LL-HH configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 75). If both checks are enabled, they are performed in the following order:

1. LL-HH
2. L-H (defined with the **L-H configuration** process parameter (page 73))

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the LL limit or HH limit attributes of the Limit definition process parameter (page 75) are set. If they are not set, the validation will fail. 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.

Duration (extended) value - Limit definition

Limits are defined as absolute values. Make sure that the limits are strictly sequential and do not overlap, so that

$$\blacksquare \quad \text{LL limit} < \text{L limit} < \text{H limit} < \text{HH limit}$$

Attribute	Type	Comment
LL limit	String	Define the names of the lower limits parameters on the DCS. Maximum length is 200 characters.
L limit	String	
H limit	String	Define the names of the upper limits parameters on the DCS. Maximum length is 200 characters.
HH limit	String	

Duration (extended) value - Limits visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows empty cells.

Attribute	Type	Comment
Enabled	Flag	Controls if the limits are visible during execution. Default setting: Yes

Duration (extended) value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt , Report , Recipe , Status change) correspond to the entries in the DCSParаметerType list. Maximum length is 200 characters.

Numeric value - Master (bundle identifier)

Specifies the parameter identifier and description of the **Numeric** data type that is to be retrieved from the DCS. The **Parameter path** indicates where the value is located on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read. Maximum length is 200 characters.
Parameter description	String	Defines an alias for the parameter. Maximum length is 200 characters.
Parameter path	String	Defines the path of the parameter. Maximum length is 400 characters.

Numeric value - L-H configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 78). If both checks are enabled, they are performed in the following order:

1. LL-HH (defined with the **LL-HH configuration** process parameter (page 77))
2. L-H

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the L limit or H limit attributes of the Limit definition process parameter (page 78) are set. If they are not set, the validation will fail. 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.

Numeric value - LL-HH configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 78). If both checks are enabled, they are performed in the following order:

1. LL-HH
2. L-H (defined with the **L-H configuration** process parameter (page 76))

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the LL limit or HH limit attributes of the Limit definition process parameter (page 78) are set. If they are not set, the validation will fail. 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.

Numeric value - Limit definition

Limits are defined as absolute values. 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)	Define the values of the lower limits (including the values themselves).
L limit	BigDecimal (Double, Float, Integer)	
H limit	BigDecimal (Double, Float, Integer)	Define the values of the upper limits (including the values themselves).
HH limit	BigDecimal (Double, Float, Integer)	

Numeric value - Limits visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows empty cells.

Attribute	Type	Comment
Enabled	Flag	Controls if the limits are visible during execution. Default setting: Yes

Numeric value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt, Report, Recipe, Status change) correspond to the entries in the DCSParameterType list. Maximum length is 200 characters.

Numeric (extended) value - Master (bundle identifier)

In an extended data type, all data is retrieved from the DCS. Thus, the **Parameter** and **Parameter for description** attributes indicate the names the data has on the DCS. The **Common path** attribute indicates where on the DCS the two named parameters and the parameters for the limits are located.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read on the DCS. Maximum length is 200 characters.
Parameter for description	String	Defines the name of the description parameter on the DCS. Maximum length is 200 characters.
Common path	String	Defines the path of all parameters of the bundle on the DCS. Maximum length is 400 characters.

Numeric (extended) value - L-H configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 81). If both checks are enabled, they are performed in the following order:

1. LL-HH (defined with the **LL-HH configuration** process parameter (page 81))
2. L-H

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the L limit or H limit attributes of the Limit definition process parameter (page 81) are set. If they are not set, the validation will fail. 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.

Numeric (extended) value - LL-HH configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 81). If both checks are enabled, they are performed in the following order:

1. LL-HH
2. L-H (defined with the **L-H configuration** process parameter (page 79))

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the LL limit or HH limit attributes of the Limit definition process parameter (page 81) are set. If they are not set, the validation will fail. 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.

Numeric (extended) value - Limit definition

Limits are defined as absolute values. 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	String	Define the names of the lower limits parameters on the DCS. Maximum length is 200 characters.
L limit	String	
H limit	String	Define the names of the upper limits parameters on the DCS. Maximum length is 200 characters.
HH limit	String	

Numeric (extended) value - Limits visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows empty cells.

Attribute	Type	Comment
Enabled	Flag	Controls if the limits are visible during execution. Default setting: Yes

Numeric (extended) value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt, Report, Recipe, Status change) correspond to the entries in the DCSParаметerType list. Maximum length is 200 characters.

String value - Master (bundle identifier)

Specifies the parameter identifier and description of the **String** data type that is to be retrieved from the DCS. The **Parameter path** indicates where the value is located on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read. Maximum length is 200 characters.
Parameter description	String	Defines an alias for the parameter. Maximum length is 200 characters.
Parameter path	String	Defines the path of the parameter. Maximum length is 400 characters.

String value - Expected value configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS must be checked against an expected value.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the Value attribute of the Expected value definition process parameter (page 83) is set. If it is not set, the validation will fail. 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.

String value - Expected value definition

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

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

String value - Expected value visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows an empty cell.

Attribute	Type	Comment
Enabled	Flag	Controls if the expected value is visible during execution. Default setting: Yes

String value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt, Report, Recipe, Status change) correspond to the entries in the DCSParameterType list. Maximum length is 200 characters.

String (extended) value - Master (bundle identifier)

In an extended data type, all data is retrieved from the DCS. Thus, the **Parameter** and **Parameter for description** attributes indicate the names the data has on the DCS. The **Common path** attribute indicates where on the DCS the two named parameters and the parameter for the expected value are located.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read on the DCS. Maximum length is 200 characters.
Parameter for description	String	Defines the name of the description parameter on the DCS. Maximum length is 200 characters.
Common path	String	Defines the path of all parameters of the bundle on the DCS. Maximum length is 400 characters.

String (extended) value - Expected value configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS must be checked against an expected value.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the Value attribute of the Expected value definition process parameter (page 85) is set. If it is not set, the validation will fail. 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.

String (extended) value - Expected value definition

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

Attribute	Type	Comment
Value	String	Defines the name of the expected value parameter on the DCS. Maximum length is 200 characters. Maximum length of the return value is 2000 characters.

String (extended) value - Expected value visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows an empty cell.

Attribute	Type	Comment
Enabled	Flag	Controls if the expected value is visible during execution. Default setting: Yes

String (extended) value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt, Report, Recipe, Status change) correspond to the entries in the DCSParаметerType list. Maximum length is 200 characters.

Timestamp value - Master (bundle identifier)

Specifies the parameter identifier and description of the **Timestamp** data type that is to be retrieved from the DCS. The **Parameter path** indicates where the value is located on the DCS.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read. Maximum length is 200 characters.
Parameter description	String	Defines an alias for the parameter. Maximum length is 200 characters.
Parameter path	String	Defines the path of the parameter. Maximum length is 400 characters.

Timestamp value - L-H configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 88). If both checks are enabled, they are performed in the following order:

1. LL-HH (defined with the **LL-HH configuration** process parameter (page 87))
2. L-H

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the L limit or H limit attributes of the Limit definition process parameter (page 88) are set. If they are not set, the validation will fail. 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.

Timestamp value - LL-HH configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 88). If both checks are enabled, they are performed in the following order:

1. LL-HH
2. L-H (defined with the **L-H configuration** process parameter (page 87))

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the LL limit or HH limit attributes of the Limit definition process parameter (page 88) are set. If they are not set, the validation will fail. 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.

Timestamp value - Limit definition

Limits are defined as absolute values. 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	Timestamp	Define the values of the lower limits (including the values themselves).
L limit	Timestamp	
H limit	Timestamp	Define the values of the upper limits (including the values themselves).
HH limit	Timestamp	

Timestamp value - Limits visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows empty cells.

Attribute	Type	Comment
Enabled	Flag	Controls if the limits are visible during execution. Default setting: Yes

Timestamp value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt, Report, Recipe, Status change) correspond to the entries in the DCSParameterType list. Maximum length is 200 characters.

Timestamp (extended) value - Master (bundle identifier)

In an extended data type, all data is retrieved from the DCS. Thus, the **Parameter** and **Parameter for description** attributes indicate the names the data has on the DCS. The **Common path** attribute indicates where on the DCS the two named parameters and the parameters for the limits are located.

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read on the DCS. Maximum length is 200 characters.
Parameter for description	String	Defines the name of the description parameter on the DCS. Maximum length is 200 characters.
Common path	String	Defines the path of all parameters of the bundle on the DCS. Maximum length is 400 characters.

Timestamp (extended) value - L-H configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 91). If both checks are enabled, they are performed in the following order:

1. LL-HH (defined with the **LL-HH configuration** process parameter (page 90))
2. L-H

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the L limit or H limit attributes of the Limit definition process parameter (page 91) are set. If they are not set, the validation will fail. 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.

Timestamp (extended) value - LL-HH configuration

Represents a system-triggered exception that is displayed in the Exception Window to define if the value retrieved from the DCS is checked against the limits defined with the **Limit definition** process parameter (page 91). If both checks are enabled, they are performed in the following order:

1. LL-HH
2. L-H (defined with the **L-H configuration** process parameter (page 89))

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the LL limit or HH limit attributes of the Limit definition process parameter (page 91) are set. If they are not set, the validation will fail. 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.

Timestamp (extended) value - Limit definition

Limits are defined as absolute values. Make sure that the limits are strictly sequential and do not overlap, so that

$$\blacksquare \quad \text{LL limit} < \text{L limit} < \text{H limit} < \text{HH limit}$$

Attribute	Type	Comment
LL limit	String	Define the names of the lower limits parameters on the DCS. Maximum length is 200 characters.
L limit	String	
H limit	String	Define the names of the upper limits parameters on the DCS. Maximum length is 200 characters.
HH limit	String	

Timestamp (extended) value - Limits visible in PEC

Specifies the display in the parameter table. If set to **No**, the system shows empty cells.

Attribute	Type	Comment
Enabled	Flag	Controls if the limits are visible during execution. Default setting: Yes

Timestamp (extended) value - Parameter type

Represents a parameter classification to allow using a parameter identifier more than once.

Attribute	Type	Comment
Parameter type	Choice list	Optional parameter to define the type of the parameter when the same parameter identifier is available for different types. The available settings (Prompt, Report, Recipe, Status change) correspond to the entries in the DCSParameterType list. Maximum length is 200 characters.

Instruction links

In addition to the permanent process parameters that are always present, the **Get DCS parameters** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
The system adds the instruction link parameter and the first link definition row to the list of parameters.
Instruction link parameters are generally inserted below all other parameters.
Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

- If no parameter is selected, the system adds the new instruction link parameter as last parameter.
- If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
- If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa. The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.
5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:



adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.



deletes the currently selected rows.



moves the currently selected row one row up.



moves the currently selected row one row down.

**REMOVING INSTRUCTION LINKS**

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction link parameter.

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

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	<p>Instruction text to be displayed.</p> <p>For any text enclosed in curly brackets you can define a hyperlink with the Instruction link definition process parameter (page 94).</p> <p>Example: Refer to {SOP1270} for guidance.</p> <p>Maximum length is 2000 characters (including HTML tags).</p>

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 94). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

`https://rockwellautomation.com`
`file:///c:/SOP/SOP1492.pdf`

Attribute	Type	Comment
Link text	Text	<p>Text to be used as link.</p> <p>For any text enclosed in curly brackets within the instruction text you can define a link with the Link URL attribute.</p> <p>Including the brackets in the link text is optional.</p> <p>Maximum length is 80 characters.</p>
Link URL	Text	<p>URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system.</p> <p>Maximum length is 256 characters.</p>

Instruction tables

In addition to the permanent process parameters that are always present, the **Get DCS parameters** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.







ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
The system adds the parameter definition and the first table row to the list of parameters.
Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction table as last parameter.
 - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
 - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
 - Select the number of columns to define the layout.
 - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
 - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:
 -  adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.
 -  deletes the currently selected rows.
 -  moves the currently selected row one row up.
 -  moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.



REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction table with its definition and rows.

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

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: 1 column , 2 columns , 3 columns , 4 columns , 5 columns . Default setting: 1 column .
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. Restriction: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Get DCS parameters** phase provides the following output variables:

Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is `true` if all parameter values have been read successfully.
 - The value is `false` if at least one of the parameter values was not read from the DCS.

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 - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is `true` if the parameter value of the boolean parameter has been read successfully.
 - The value is `false` if the parameter value of the boolean parameter was not read from the DCS.

Boolean value - Value

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

Boolean (extended) value

If you have added Boolean (extended) 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 (extended) value bundles:

Boolean (extended) value - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is `true` if the parameter values of the boolean parameter have been read successfully.
 - The value is `false` if a parameter value of the boolean parameter (value, description, expected value) was not read from the DCS.

Boolean (extended) value - Value

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

Duration value

If you have added Duration 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 Duration value bundles:

Duration value - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is **true** if the parameter value of the duration parameter has been read successfully.
 - The value is **false** if the parameter value of the duration parameter was not read from the DCS.

Duration value - Value

- Data type: Duration, used for displaying time spans and for time-related calculations.
- Usage: The output variable provides the value of the duration parameter. The value is Null if N/A is the phase result.

Duration (extended) value

If you have added Duration (extended) 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 Duration (extended) value bundles:

Duration (extended) value - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is `true` if the parameter values of the duration parameter have been read successfully.
 - The value is `false` if a parameter value of the duration parameter (value, description, limit) was not read from the DCS.

Duration (extended) value - Value

- Data type: Duration, used for displaying time spans and for time-related calculations.
- Usage: The output variable provides the value of the duration parameter. 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 - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is `true` if the parameter value of the numeric parameter has been read successfully.
 - The value is `false` if the parameter value of the numeric parameter was not read from the DCS.

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 parameter as a **BigDecimal** value. The value is `Null` if N/A is the phase result.

Numeric (extended) value

If you have added Numeric (extended) 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 (extended) value bundles:

Numeric (extended) value - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is `true` if the parameter values of the numeric parameter have been read successfully.
 - The value is `false` if a parameter value of the numeric parameter (value, description, limit) was not read from the DCS.

Numeric (extended) 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 parameter 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 - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is `true` if the parameter value of the string parameter has been read successfully.
 - The value is `false` if the parameter value of the string parameter was not read from the DCS. respective user-triggered exception.

String value - Value

- 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 value of the string parameter. The value is Null if N/A is the phase result.

String (extended) value

If you have added String (extended) 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 (extended) value bundles:

String (extended) value - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is `true` if the parameter values of the string parameter have been read successfully.
 - The value is `false` if a parameter value of the string parameter (value, description, expected value) was not read from the DCS.

String (extended) value - Value

- 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 value of the string parameter. The value is Null if N/A is the phase result.

Timestamp value

If you have added Timestamp 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 Timestamp value bundles:

Timestamp value - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is `true` if the parameter value of the timestamp parameter has been read successfully.
 - The value is `false` if the parameter value of the timestamp parameter was not read from the DCS.

Timestamp value - Value

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
- Usage: The output variable provides the value of the timestamp parameter. The value is Null if N/A is the phase result.

Timestamp (extended) value

If you have added Timestamp (extended) 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 Timestamp (extended) value bundles:

Timestamp (extended) value - Retrieval successful

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the get operation from the DCS was successful.
 - The value is **true** if the parameter values of the timestamp parameter have been read successfully.
 - The value is **false** if a parameter value of the timestamp parameter (value, description, limit) was not read from the DCS.

Timestamp (extended) value - Value

- Data type: Timestamp, used for displaying dates and times and for time-related calculations.
- Usage: The output variable provides the value of the timestamp parameter. 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".

Get DCS Alarms

The **Get DCS alarms** phase allows an operator to request alarm-specific data from a batch running on a DCS.

It can be used for processing requirements, such as:

- Concurrent retrieval of alarms for recording in the batch report
Alarms on the DCS are retrieved every five minutes. In case an alarm has occurred, the alarm can be converted into an exception. Then it is documented in the batch report and included in the review and approval process.
- Retrieval of alarms after completion of a batch run for recording in the batch report
After a batch run has been completed, all alarms that have occurred during the run are retrieved and converted into exceptions. They are documented in the batch report and included in the review and approval process.

Execution

In addition to the instruction text, the **Get DCS alarms** phase displays the following data:

- the filter criteria applied when the system retrieves alarms from the DCS.
 - DSC name
 - Start and end of the retrieval period
 - Batch ID
 - Unit ID
 - Modules
- a table that lists all alarms that have been retrieved and have not yet been converted into exceptions. For each retrieved alarm, the system displays the following information:
 - the timestamp when the alarm occurred,
 - the source of the alarm,
 - the alarm text as created by the DCS,
 - an alarm comment, if a comment has already been added in the DCS.

- an information message shows the status of the retrieval:
 - **Time of last update unknown** indicates that no retrieval has been performed yet,
 - **Update in progress** indicates that the operator has tapped the **Get** button and the retrieval process is ongoing,
 - **Last update completed (<timestamp>)** indicates that the last retrieval was successful and when it took place,
 - **Last update failed (<timestamp>)** indicates that the last retrieval encountered an issue and when the issue occurred.

When the operator taps the **Get** button, the phase accesses the DCS and retrieves all alarms that match the filter criteria. The operator can tap the button multiple times in order to retrieve any further matching alarms that have occurred since the last **Get** operation.

If the phase is set to retrieve alarms automatically (page 112), it accesses the DCS and queries for alarms as soon as it becomes active. Afterwards, it continues to query the DCS at the defined interval until the operator taps the **Confirm** button to complete the phase. Alarms are sorted by their timestamps in descending order. This means that newly retrieved alarms always appear at the top of the list of alarms.

TIP

Please note that when the system retrieves and lists new alarms, it considers all unit procedures of the order or workflow and only displays those alarms that have not yet been processed by any other phase or phase run.

Once the phase has retrieved alarms from the DCS, it displays them in the table with one row per alarm. The operator can tap to select individual alarm rows or select all alarms at once by tapping the checkbox in the table header. Selecting an alarm enables the **Exception** button to convert the alarm into an exception (page 114). The exception indicates the number of converted alarms and shows the alarm details received from the DCS as comment, one comment per alarm. Alarms that have been converted are removed from the table.

The phase expects all alarms to be converted into exceptions. If the operator tries to complete the phase when there are still alarms listed in the table, he can only do this by recording an exception for the unconverted alarms (page 113).

TIP

The system can also be configured to allow completing the phase without having converted the retrieved alarms into exceptions.

Tapping the **Confirm** button always triggers another alarms retrieval to ensure that the operator can only complete the phase if there are no unprocessed alarms.

After completion the phase displays the number of retrieved alarms in the Execution Window.

The Navigator displays the number of retrieved alarms.

Timestamp	Source	Alarm	Comment
11/09/2017 01:12:21 PM CET		Attr: LO_ALM Word: LOW Level: 10-WARNING Desc: Low Alarm Value 64.79 Limit 65	
11/09/2017 01:12:08 PM CET		Attr: LO_ALM Word: LOW Level: 10-WARNING Desc: Low Alarm Value 64.11 Limit 65	
11/09/2017 01:11:50 PM CET		Attr: LO_ALM Word: LOW Level: 15-CRITICAL Desc: Low Alarm Value 61.37 Limit 65	Adjusted pressure at valve7
11/09/2017 01:10:40 PM CET		Attr: LO_ALM Word: LOW Level: 15-CRITICAL Desc: Low Alarm Value 60.46 Limit 65	Adjusted pressure at valve5
11/09/2017 12:49:52 PM CET		Attr: LO_ALM Word: LOW Level: 15-CRITICAL Desc: Low Alarm Value 60.46 Limit 65	Adjusted pressure at valve5

Figure 21: Get DCS alarms during execution

Timestamp	Source	Alarm	Comment
11/09/2017 01:12:21 PM CET		Attr: LO_ALM Word: LOW Level: 10-WARNING Desc: Low Alarm Value 64.79 Limit 65	
11/09/2017 01:12:08 PM CET		Attr: LO_ALM Word: LOW Level: 10-WARNING Desc: Low Alarm Value 64.11 Limit 65	
11/09/2017 01:11:50 PM CET		Attr: LO_ALM Word: LOW Level: 15-CRITICAL Desc: Low Alarm Value 61.37 Limit 65	Adjusted pressure at valve7
11/09/2017 01:10:40 PM CET		Attr: LO_ALM Word: LOW Level: 15-CRITICAL Desc: Low Alarm Value 60.46 Limit 65	Adjusted pressure at valve5
11/09/2017 12:49:52 PM CET		Attr: LO_ALM Word: LOW Level: 15-CRITICAL Desc: Low Alarm Value 60.46 Limit 65	Adjusted pressure at valve5

Figure 22: Get DCS alarms after phase completion

Figure 23: Get DCS alarms in the Navigator

Phase Design

The characteristics of the **Get DCS 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. The right column of the first row shows the timestamp of the query start and below that, in the second row, the query end timestamp.

The second, third, and fourth rows of the left column contain the remaining filter criteria, as well as the third row of the center column.

In the fourth row of the right column, the phase provides the **Get** and **Exception** buttons. The next rows display a tabular view of retrieved alarms and their data, spanning all columns.

The merged left and center columns of the bottom row display the information message on the status of the alarms retrieval.

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

When the phase is completed it shows the same three-column, multi-row layout, however without the **Get** and **Exception** buttons and a text row with the number of retrieved alarms instead of 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).

DCS

Defines the name of the DCS system to be accessed for alarms retrieval.

Attribute	Type	Comment
Name	String	Logical name of the DCS to be used. The available entries correspond to the entries in the DCSNames list. Default setting: First entry in the list. Maximum length is 255 characters.

Filter criteria

Defines the filter that is applied to the alarms that occur on the selected DCS.

Attribute	Type	Comment
Batch ID	String	Optional parameter to define the identifier of the batch running on the DCS. Maximum length is 250 characters. If not defined, phase displays "N/A".
Unit ID	String	Optional parameter to define the identifier of a unit of the DCS. Maximum length is 250 characters. If not defined, phase displays "N/A".
Query start	Timestamp	Optional parameter to set the start timestamp from which on the alarms are queried. If not defined, phase displays "N/A".
Query end	Timestamp	Optional parameter to set the end timestamp up to which the alarms are queried. If not defined, phase displays "N/A".
Module IDs	Text (structured)	Optional parameter to define the list of equipment module IDs and control module IDs. Maximum length is 2000 characters. If not defined, phase displays "N/A".

For compiling the list of module identifiers, the system provides a List editor.

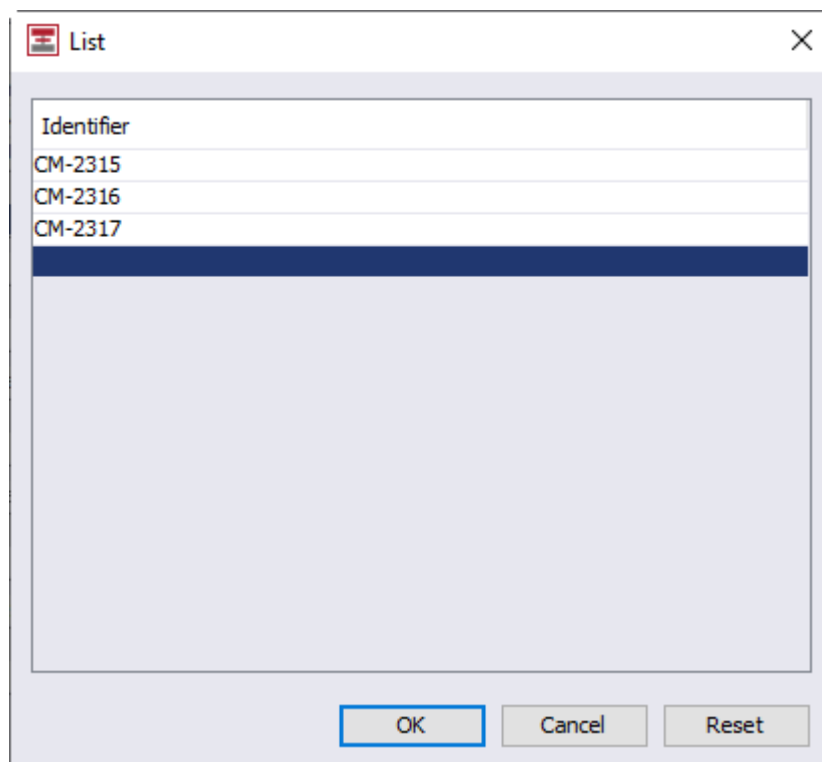


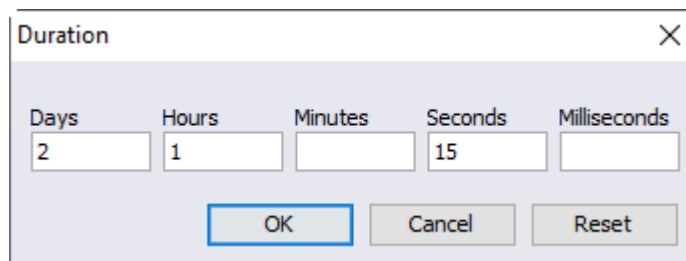
Figure 24: List editor

Automatic update

Defines if the phase accesses the DCS automatically at a given interval without further operator interaction.

Attribute	Type	Comment
Enabled	Boolean	Controls if the query is repeated automatically. If so, make sure to define the Update interval attribute.
Update interval	Duration	Defines the interval between re-querying. The minimum interval is set to 1 minute if the interval is not defined at all or configured to be less than that.

For defining the interval, the system provides a Duration editor.



The image shows a 'Duration' dialog box with a close button (X) in the top right corner. It contains five input fields for time units: Days (value 2), Hours (value 1), Minutes (empty), Seconds (value 15), and Milliseconds (empty). At the bottom, there are three buttons: 'OK' (highlighted with a blue border), 'Cancel', and 'Reset'.

Figure 25: Duration editor

Retrieval exception

Represents a system-triggered exception that is displayed in the Exception Window to record incidents when the DCS cannot be accessed for retrieving alarms.

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.

Unconverted alarms

Represents a system-triggered exception that is displayed in the Exception Window to record incidents when one or more alarms that were retrieved from the DCS are not converted into exceptions before the phase is completed.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. Default setting: Yes

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.

Record alarm exception

Represents a user-triggered exception that is accessed via the **Exception** button above the table of alarms in the Execution Window.

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.

Instruction links

In addition to the permanent process parameters that are always present, the **Get DCS alarms** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
The system adds the instruction link parameter and the first link definition row to the list of parameters.
Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
 - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
 - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa. The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

- Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:



adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.



deletes the currently selected rows.



moves the currently selected row one row up.



moves the currently selected row one row down.



REMOVING INSTRUCTION LINKS

- In the list of parameters, select the instruction link parameter you wish to remove.
- Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction link parameter.

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

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	<p>Instruction text to be displayed.</p> <p>For any text enclosed in curly brackets you can define a hyperlink with the Instruction link definition process parameter (page 117).</p> <p>Example: Refer to {SOP1270} for guidance.</p> <p>Maximum length is 2000 characters (including HTML tags).</p>

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 116). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

`https://rockwellautomation.com`

`file:///c:/SOP/SOP1492.pdf`

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the Link URL attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Get DCS alarms** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.




ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
 2. Select the **Instruction Table** type.
The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
 3. Type an identifier and click the **OK** button.
The system adds the parameter definition and the first table row to the list of parameters.
Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction table as last parameter.
 - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
 - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.
- TIPS**


Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.


The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.
4. Specify the overall appearance of the table:
 - Select the number of columns to define the layout.
 - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
 - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.

5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

 adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

 deletes the currently selected rows.

 moves the currently selected row one row up.

 moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.



REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction table with its definition and rows.

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

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: 1 column , 2 columns , 3 columns , 4 columns , 5 columns . Default setting: 1 column .

Attribute	Type	Comment
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. Restriction: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

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

Number of retrieved alarms

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the number of alarms that were retrieved by the phase.

Number of converted alarms

- Data type: Long, used for integral numbers:
12345
- Usage: The output variable provides the number of alarms that were converted into exceptions.

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".

DCS Alarm-based Trigger

The **DCS alarm-based trigger** phase allows to automatically create runs of an event-triggered operation (ETO) based on DCS alarms whenever new DCS alarms are retrieved in the defined check cycle.

It can be used for processing requirements, such as:

- Several operators are responsible for a production area in which several orders are executed. The operators have to be notified when new alarms have been retrieved from the DCS. Subsequently, an operator can document the alarms as PharmaSuite exceptions.

Execution

The **DCS alarm-based trigger** phase is intended for being run on the Operation Execution (OE) server. It generates events to trigger the runs of event-triggered operations, such as operations that check for alarms that have occurred on a connected DCS.

As a server-run phase it starts automatically and becomes active when an operator starts an operation that references the phase as its trigger phase. Once active it generates trigger events according to its configured schedule.

The phase completes automatically when the last operation whose runs it triggers is completed by an operator.

Phase Design

The characteristics of the **DCS alarm-based trigger** phase are defined via process parameters and their attributes.

Since it runs on the OE server and does not require operator interaction, it is not visible to operators during processing.

When the phase is completed, the batch report shows the configuration of the alarm query, as well as the number of alarms it retrieved and how many triggers it sent while it was active.

Process Parameters

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

DCS

Defines the name of the DCS to be accessed for alarms retrieval.

TIP

Make sure the DCS you select is identical with the DCS selected for the **Get DCS alarms** phase of the operation whose **Trigger-enabled** capability references the trigger phase.

Attribute	Type	Comment
Name	String	Logical name of the DCS to be used. The available entries correspond to the entries in the DCSNames list. Default setting: First entry in the list. Maximum length is 255 characters.

Filter criteria

Defines the filter that is applied to the alarms that occur on the selected DCS.

TIP

Make sure the filter criteria you define are identical with those defined for the **Get DCS alarms** phase of the operation whose **Trigger-enabled** capability references the trigger phase.

Attribute	Type	Comment
Batch ID	String	Optional parameter to define the identifier of the batch running on the DCS. Maximum length is 250 characters. If not defined, phase displays "N/A".
Unit ID	String	Optional parameter to define the identifier of a unit of the DCS. Maximum length is 250 characters. If not defined, phase displays "N/A".
Query start	Timestamp	Optional parameter to set the start timestamp from which on the alarms are queried. If not defined, phase displays "N/A".

Attribute	Type	Comment
Query end	Timestamp	Optional parameter to set the end timestamp up to which the alarms are queried. If not defined, phase displays "N/A".
Module IDs	Text (structured)	Optional parameter to define the list of equipment module IDs and control module IDs. Maximum length is 2000 characters. If not defined, phase displays "N/A".

For compiling the list of module identifiers, the system provides a List editor.

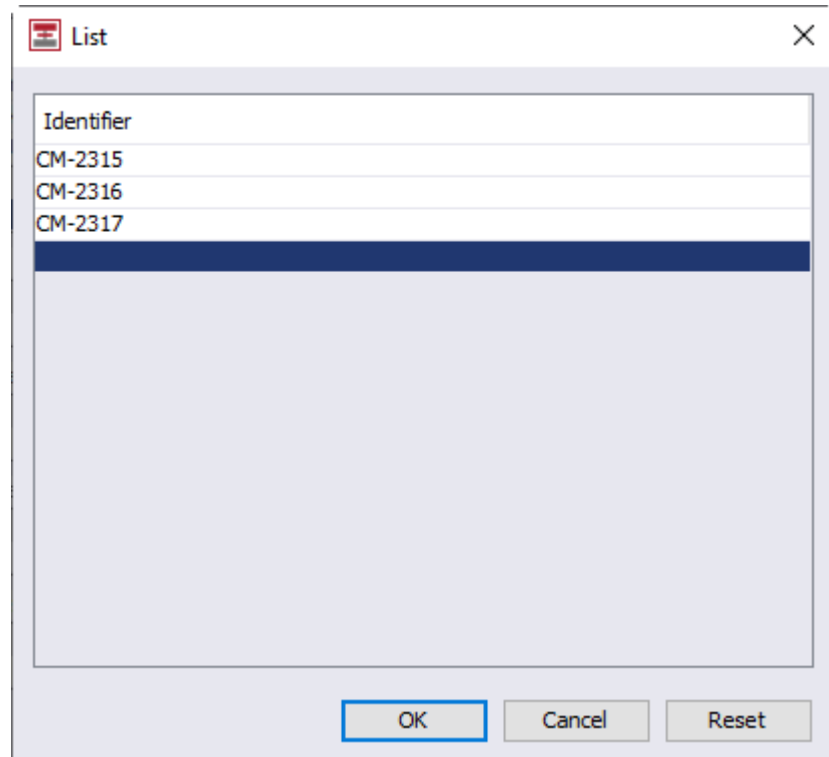


Figure 26: List editor

Retrieving cycle

Defines the interval between two consecutive reading actions.

Attribute	Type	Comment
Duration	Duration	The minimum interval is one minute. So if you set a value of less than one minute or leave the cell blank, the system interprets this as one minute.

For defining the interval, the system provides a Duration editor.

Figure 27: Duration editor

Timeout period

Defines how long the phase waits for its first event-triggered operation to be started before it completes automatically. This prevents deadlocks that might occur if an event-triggered operation fails to start entirely.

Attribute	Type	Comment
Duration	Duration	If left blank, the system interprets this as 30 minutes.

For defining the period, the system provides a Duration editor.

Figure 28: Duration editor

Timeout exception

Represents a system-triggered exception that is displayed in the overview Exception Window and in the batch report.

The system records a timeout incident as exception, since the quality of a product may be compromised if scheduled events, such as reading alarms, do not take place.

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.

Retrieval exception

Represents a system-triggered exception that is displayed in the overview Exception Window and in the batch report.

The system records a failed retrieval incident as exception, since the quality of a product may be compromised if scheduled events, such as reading alarms, cannot be executed.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception. Since there is no operator interaction for the exception, it is not linked to a 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.

Post-ETO alarms exception

Represents a system-triggered exception that is displayed in the overview Exception Window and in the batch report.

The system records an exception when new alarms have occurred in the DCS after the ETO template was removed and before the corresponding **Get DCS alarms** phase was completed, since this may compromise product quality.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception. Since there is no operator interaction for the exception, it is not linked to a 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 operation or the calculated result of several outputs as value into a parameter attribute.

When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **DCS alarm-based trigger** 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"**.

Set Order Context

The **Set order context** phase allows an operator to set an order context on a DCS.

It can be used for processing requirements, such as:

- Providing an order context for processing the order on an automation system
Based on an existing master recipe, batch, and order, the phase allows to process the order on a DCS.

Execution

In addition to the instruction text, the **Set order context** phase displays the following data:

- the basic data of the order whose context is to be set on the DCS:
 - DSC name
 - Batch ID
 - Master recipe ID
 - Order ID
 - Material ID
- a table that lists all MFC positions of the order. For each MFC position, the system displays the following information:
 - Material ID
 - Quantity with unit of measure
 - Batches allocated to the MFC position
 - Order step, in which the MFC position is used
 - MFC type of the position (**Process input**, **Process output**)
- an information message shows the status of the set operation:
 - **Context not set yet** indicates that no set operation has been performed yet,
 - **Context set successfully** (<timestamp>) indicates that the order context has successfully been transferred to the DCS,
 - **Context set manually** (<timestamp>) indicates that the context was not transferred but set manually on the DCS,

- **Setting context failed** indicates that the order context could not be transferred successfully.

When the operator taps the **Set** button, the phase accesses the DCS and transfers the order data and its MFC positions. The **Set** button is disabled after the order context data has been sent regardless of the operation's success. Any further set operations are considered exceptions.

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

- In the **Manual completion** mode, the operator manually triggers to set the order context.
- In the **Automatic completion** mode, the phase sets the order context and is completed automatically without any operator interaction.


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

- to re-send the order context to the DCS,
- to confirm that the order context has been set manually on the DCS.

After completion the phase displays the order with its data in the Execution Window. The Navigator displays the name of the current unit procedure.

Set the order context on the DCS.

DCS name: JavaDCSMock Batch ID: BX399
Master recipe ID: ID_SRTD-001 [3] Order ID: ID-SRTD-005
Material ID: D130-01

Set 

MFC pos.	Material ID	Quantity	UoM	Allocated batches	Order step	MFC type
140	D003-04	10.0 g			ID-SRTD-003-Automated Tableting Run	Process input
150	D005-04	20.0 g			ID-SRTD-003-Automated Tableting Run	Process input
160	D130-01	1,000.0 g			ID-SRTD-003-Automated Tableting Run	Process input
170	D130-01	1,010.0 g			ID-SRTD-003-Automated Tableting Run	Process output

Context not set yet


Confirm 

Figure 29: Set order context during execution

Re-send the order context.

Confirm 

Set the order context manually.

Confirm 

Figure 30: User-triggered exceptions of Set order context

Set the order context on the DCS.

DCS name: JavaDCSMock Batch ID: BX399
 Master recipe ID: ID_SRTD-001 [3] Order ID: ID-SRTD-005
 Material ID: D130-01

Set ✓

MFC pos.	Material ID	Quantity	UoM	Allocated batches	Order step	MFC type
140	D003-04	10.0 g			ID-SRTD-005-Automated Tableting Run	Process input
150	D005-04	20.0 g			ID-SRTD-005-Automated Tableting Run	Process input
160	D130-01	100.0 g			ID-SRTD-005-Automated Tableting Run	Process input
170	D130-01	1,010.0 g			ID-SRTD-005-Automated Tableting Run	Process output

Context set successfully (01/10/2018 11:52:01 AM CET)

Confirm ✓

Figure 31: Set order context after phase completion

Stand-alone DCS Run
Set Order Context

ID-SRTD-005

Figure 32: Set order context in the Navigator

Phase Design

The characteristics of the **Set order context** 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, third, and fourth rows of the left and center columns contain the basic data of the order context.

In the fourth row of the right column, the phase provides the **Set** button.

The next rows display a tabular view of the MFC positions of the order, spanning all columns.

The merged left and center columns of the bottom row display the information message on the set operation.

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

When the phase is completed it shows the same three-column, multi-row layout.

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 is automatically completed after the order context has been set successfully on the DCS.

DCS

Defines the name of the DCS system to be accessed for setting the order context.

Attribute	Type	Comment
Name	String	Logical name of the DCS to be used. The available entries correspond to the entries in the DCSNames list. Default setting: First entry in the list. Maximum length is 255 characters.

Set order context manually

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to confirm that the order context has been set manually at the DCS.

It covers incidents when setting the order context has failed and cannot be repeated.

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.

Re-send order context

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to re-send the order context to the DCS.

It covers incidents when setting the order context has 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.

Instruction links

In addition to the permanent process parameters that are always present, the **Set order context** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
The system adds the instruction link parameter and the first link definition row to the list of parameters.
Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
 - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
 - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa. The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

- Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:



adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.



deletes the currently selected rows.



moves the currently selected row one row up.



moves the currently selected row one row down.



REMOVING INSTRUCTION LINKS

- In the list of parameters, select the instruction link parameter you wish to remove.
- Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction link parameter.

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

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	<p>Instruction text to be displayed.</p> <p>For any text enclosed in curly brackets you can define a hyperlink with the Instruction link definition process parameter (page 138).</p> <p>Example: Refer to {SOP1270} for guidance.</p> <p>Maximum length is 2000 characters (including HTML tags).</p>

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 137). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

`https://rockwellautomation.com`

`file:///c:/SOP/SOP1492.pdf`

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the Link URL attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Set order context** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
The system adds the parameter definition and the first table row to the list of parameters.
Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction table as last parameter.
 - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
 - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
 - Select the number of columns to define the layout.
 - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
 - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.

- Specify the table rows and their content. The button bar above the rows table provides the following functions:



adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.



deletes the currently selected rows.



moves the currently selected row one row up.



moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.



REMOVING INSTRUCTION TABLES

- In the list of parameters, select the instruction table parameter you wish to remove.
- Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction table with its definition and rows.

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

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: 1 column, 2 columns, 3 columns, 4 columns, 5 columns. Default setting: 1 column.

Attribute	Type	Comment
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. Restriction: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Set order context** 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".

-
-
- FT PharmaSuite® BB - User Manual DCS Phases
-
-

Show Consumed Material

The **Show consumed material** phase allows an operator to display material consumption data provided by a DCS.

It can be used for processing requirements, such as:

- Including data on material that was consumed on an automation system in the batch report created by the MES.

Execution

The **Show consumed material** phase displays the list of MFC inputs of the order with their planned and recorded quantities. As it receives data on the consumed material from the DCS, it adds sub-rows for the individual material positions. For each consumed subplot, the subplot-specific rows show

- the batch and subplot ID,
- the quantity recorded as consumed,
- the time when the consumption happened on the DCS and who performed it,
- the sender of the data as recorded on the DCS,
- whether the subplot was fully consumed in the process.

When the phase becomes active, it shows all consumptions that have been received from the DCS so far. The table does not update automatically to reflect any new consumptions. Instead the phase provides a **Refresh** button to update the table with all consumption data that has been received from the DCS in the meantime.


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

- to remove an incorrectly recorded consumption from the list,
- to add a missing consumption to the list.

After completion the phase displays the consumption-related data of a material with its data in the Execution Window.

The Navigator displays the number of consumption entries.

Confirm consumption data from the DCS.

Refresh 

MFC pos.	Material	Batch ID / sublot ID	Planned	Recorded	DCS execution timestamp	DCS performer	Sender	Sublot consumed
140	D003-04 / Ethanol 96 %	N/A	10.0 g	10.0 g	01/10/2018 11:59:46 AM CET	Raita, Oupah (oraita)	MES	<input checked="" type="checkbox"/>
150	D005-04 / Talc	N/A	20.0 g	0.0 g				
160	D130-01 / Sonolin 100 mg premix	N/A	100.0 g	0.0 g				


Confirm 

Figure 33: Show consumed material during execution

Select the incorrectly recorded data set to be removed.

MFC pos.	Material	Batch ID / sublot ID	Planned	Recorded	DCS execution timestamp	DCS performer	Sender	Sublot consumed
140	D003-04 / Ethanol 96 %	N/A	10.0 g	10.0 g	01/10/2018 11:59:46 AM CET	Raita, Oupah (oraita)	MES	<input type="checkbox"/>
		BX366 / SL00000314		10.0 g				<input checked="" type="checkbox"/>
150	D005-04 / Talc	N/A	20.0 g	0.0 g				<input type="checkbox"/>
160	D130-01 / Sonolin 100 mg premix	N/A	100.0 g	0.0 g				<input type="checkbox"/>

Confirm 

Add a consumption for the selected material.

MFC position	Material ID	Material short description	Unit of measure
140	D003-04	Ethanol 96 %	g
150	D005-04	Talc	g
160	D130-01	Sonolin 100 mg premix	g

Batch ID Sublot ID


Timestamp Performer

Consumed qty. Consumed completely ☐ Yes ☐ No ☐

Confirm 

Figure 34: User-triggered exceptions of Show consumed material

Confirm consumption data from the DCS.

Refresh 

MFC pos.	Material	Batch ID / sublot ID	Planned	Recorded	DCS execution timestamp	DCS performer	Sender	Sublot consumed
140	D003-04 / Ethanol 96 %	N/A	10.0 g	10.0 g	01/10/2018 11:59:46 AM CET	Raita, Oupah (oraita)	MES	<input checked="" type="checkbox"/>
		BX366 / SL00000314		10.0 g				
150	D005-04 / Talc	N/A	20.0 g	20.0 g	01/10/2018 12:05:06 PM CET	Raita, Oupah (oraita)	MES	<input checked="" type="checkbox"/>
		BX367 / SL00000327		20.0 g				
160	D130-01 / Sonolin 100 mg premix	N/A	100.0 g	1,000.0 g	01/10/2018 12:06:33 PM CET	Raita, Oupah (oraita)	MES	<input checked="" type="checkbox"/>
		BX368 / SL00000338		1,000.0 g				


Confirm 

Figure 35: Show consumed material after phase completion

Stand-alone DCS Run Show Consumed Material

3 consumptions

Figure 36: Show consumed material in the Navigator

Phase Design

The characteristics of the **Show consumed material** 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 first and second columns provide space for textual instructions. The right column contains the **Refresh** button.

The next rows display a tabular view of the MFC inputs of the order, spanning all columns.

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

When the phase is completed it shows the same three-column, multi-row layout.

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).

Add consumption

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to add a consumption data set that is missing from the list.

It covers incidents when a consumption was performed on the DCS, but was not received by the system due to a communication issue.

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.

Remove incorrectly recorded data set

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to remove a consumption data set of a subplot that was not recorded correctly.

It covers incidents when a consumption data set was received more than once by the system due to a communication issue.

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.

Instruction links

In addition to the permanent process parameters that are always present, the **Show consumed material** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
The system adds the instruction link parameter and the first link definition row to the list of parameters.
Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
 - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
 - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa. The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

- Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:



adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.



deletes the currently selected rows.



moves the currently selected row one row up.



moves the currently selected row one row down.



REMOVING INSTRUCTION LINKS

- In the list of parameters, select the instruction link parameter you wish to remove.
- Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction link parameter.

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

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Instruction text	Type	Comment
Instruction text	HTML text	<p>Instruction text to be displayed.</p> <p>For any text enclosed in curly brackets you can define a hyperlink with the Instruction link definition process parameter (page 151).</p> <p>Example: Refer to {SOP1270} for guidance.</p> <p>Maximum length is 2000 characters (including HTML tags).</p>

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 150). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

`https://rockwellautomation.com`

`file:///c:/SOP/SOP1492.pdf`

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the Link URL attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Show consumed material** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION TABLES


1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
The system adds the parameter definition and the first table row to the list of parameters.
Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction table as last parameter.
 - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
 - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.
4. Specify the overall appearance of the table:
 - Select the number of columns to define the layout.
 - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
 - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.


TIPS


Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.


The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

5. Specify the table rows and their content. The button bar above the rows table provides the following functions:

 adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

 deletes the currently selected rows.

 moves the currently selected row one row up.

 moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.



REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction table with its definition and rows.

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

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: 1 column, 2 columns, 3 columns, 4 columns, 5 columns . Default setting: 1 column .

Attribute	Type	Comment
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. Restriction: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Show consumed material** 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"**.

Show Produced Material

The **Show produced material** phase allows an operator to display material production data provided by a DCS.

It can be used for processing requirements, such as:

- Including data on material that was produced on an automation system in the batch report created by the MES.

Execution

In addition to the instruction text, the **Show produced material** phase displays the following data:

- the data of the MFC output that is being produced on the DCS
 - Material identifier and short description
 - MFC position
 - Batch ID
 - Planned quantity
 - Total recorded quantity
- the list of sublots produced on the DCS with the following data for each subplot
 - Sublot ID
 - Recorded quantity
 - Time when the production happened on the DCS and who performed it
 - The sender of the data as recorded on the DCS
- As it receives data on the produced material from the DCS, it adds sub-rows for the individual sublots.

When the phase becomes active, it shows all production data that has been received from the DCS so far. The table does not update automatically to reflect any new produced sublots. Instead the phase provides a **Refresh** button to update the table with all production data that has been received from the DCS in the meantime.

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


- to either add a new subplot with its required data or
- to correct the quantity of an existing subplot by typing into the **Quantity** box the amount by which the quantity needs to be corrected. To reduce the subplot quantity, the operator needs to enter a negative number. To increase the subplot quantity, the operator needs to enter a positive number.

After completion the phase displays the production-related data of a material with its data in the Execution Window.

The Navigator displays the number of production entries.

Confirm production data from the DCS.

Material: D130-01 / Sonolin 100 mg premix
MFC position: 170 Batch ID: BX399
Planned quantity: 1,010.0 g Total recorded quantity: 1,010.0 g

Refresh 

Sublot ID	Recorded	DCS execution timestamp	DCS performer	Sender
SL00000485	1,010.0 g	01/10/2018 12:08:54 PM CET	Raita, Oupah (oraita)	MES


Confirm 

Figure 37: Show produced material during execution

Add new or correct produced material.

Material: D130-01 / Sonolin 100 mg premix
Batch ID: N/A

Sublot ID Quantity
Timestamp Performer



Confirm 

Figure 38: User-triggered exception of Show produced material

Confirm production data from the DCS.

Material: D130-01 / Sonolin 100 mg premix
MFC position: 170 Batch ID: BX399
Planned quantity: 1,010.0 g Total recorded quantity: 1,010.0 g

Refresh 

Sublot ID	Recorded	DCS execution timestamp	DCS performer	Sender
SL00000485	1,010.0 g	01/10/2018 12:08:54 PM CET	Raita, Oupah (oraita)	MES


Confirm 

Figure 39: Show produced material after phase completion

Stand-alone DCS Run
Show Produced Material

1 production

Figure 40: Show produced material in the Navigator

Phase Design

The characteristics of the **Show produced material** 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, third, and fourth rows of the left and center columns contain the basic data of the MFC output.

In the fourth row of the right column, the phase provides the **Refresh** button.

The next rows display a tabular view of the sublots produced for the order, spanning all columns.

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

When the phase is completed it shows the same three-column, multi-row layout.

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).

Correct produced material

Represents a user-triggered exception that is accessible from the Exception Window. The exception allows an operator to add a new produced subplot or to correct the quantity of an existing produced subplot. It covers incidents when the production data set received from the DCS does not correspond to the actual subplot 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.

Instruction links

In addition to the permanent process parameters that are always present, the **Show produced material** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.







ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.
The system adds the instruction link parameter and the first link definition row to the list of parameters.
Instruction link parameters are generally inserted below all other parameters.
Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
 - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
 - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa. The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.
5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:
 -  adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.
 -  deletes the currently selected rows.
 -  moves the currently selected row one row up.
 -  moves the currently selected row one row down.

**REMOVING INSTRUCTION LINKS**

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction link parameter.

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

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	<p>Instruction text to be displayed.</p> <p>For any text enclosed in curly brackets you can define a hyperlink with the Instruction link definition process parameter (page 162).</p> <p>Example: Refer to {SOP1270} for guidance.</p> <p>Maximum length is 2000 characters (including HTML tags).</p>

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 162). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

`https://rockwellautomation.com`

`file:///c:/SOP/SOP1492.pdf`

Attribute	Type	Comment
Link text	Text	<p>Text to be used as link.</p> <p>For any text enclosed in curly brackets within the instruction text you can define a link with the Link URL attribute.</p> <p>Including the brackets in the link text is optional.</p> <p>Maximum length is 80 characters.</p>
Link URL	Text	<p>URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system.</p> <p>Maximum length is 256 characters.</p>

Instruction tables

In addition to the permanent process parameters that are always present, the **Show produced material** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.







ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
The system adds the parameter definition and the first table row to the list of parameters.
Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction table as last parameter.
 - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
 - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
 - Select the number of columns to define the layout.
 - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
 - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:
 -  adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.
 -  deletes the currently selected rows.
 -  moves the currently selected row one row up.
 -  moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.



REMOVING INSTRUCTION TABLES

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.

The system asks you to confirm the action and then removes the instruction table with its definition and rows.

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

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: 1 column , 2 columns , 3 columns , 4 columns , 5 columns . Default setting: 1 column .
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. Restriction: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Show produced material** 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".

-
-
- FT PharmaSuite® BB - User Manual DCS Phases
-
-

Wait for Event (OES)

The **Wait for event (OES)** phase waits for an event from a DCS and allows to automatically synchronize an operation that runs on the OE server and the DCS or an operator-run operation.

It can be used for processing requirements, such as:

- Receive a message from an automation system
A process has reached a certain stage on the automation system and the MES can continue with this process.
- Remind the operator of a task
After the operator has verified that a manufacturing step was completed, the operator must receive a reminder after a defined period of time to proceed with a further procedural step.
This example makes use of the **Time-based trigger** phase, an IPC phase.

Execution

The **Wait for event (OES)** phase is intended for being run on the Operation Execution (OE) server. It waits for an event message from the DCS or a **Send event** phase (page 187). As a server-run phase it starts automatically when an operator starts its operation and completes automatically when it receives the expected message or an abort message.

TIP

Please note that a server-run phase can only be aborted with a corresponding message from a **Send event** phase. For this reason it is essential to include a **Send event** phase configured to send an abort broadcast in the same unit procedure as the **Wait for event (OES)** phase. Otherwise, an issue that prevents the expected event message from being received by the **Wait for event (OES)** phase would lead to a processing deadlock.

Phase Design

The characteristics of the **Wait for event (OES)** phase are defined via process parameters and their attributes.

Since it runs on the OE server and does not require operator interaction, it is not visible to operators during processing.

When the phase is completed, the batch report shows the identifier and timestamp of the received message as well as whether or not the phase was aborted.

Process Parameters

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

Event message

Indicates the identifier of the message to be received from the DCS or a **Send event** phase.

It needs to correspond to the identifier defined in the **Event message** process parameter (page 190) of the **Send event** phase and must always be defined. Otherwise the phase cannot complete without exception.

Attribute	Type	Comment
Identifier	String	Identifier of the message to be received from the DCS or the Send event phase. Maximum length is 80 characters.

Abort wait

Represents a system-triggered exception that is displayed in the overview Exception Window and in the batch report.

The system records it as exception when the phase was aborted, since this may interfere with the intended processing sequence.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception. Since there is no operator interaction for the exception, it is not linked to a 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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Wait for event** phase provides the following output variables:

Wait aborted

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the wait state was aborted.
 - The value is `true` if the phase was completed with an **Abort wait** exception.
 - The value is `false` if the phase was completed after having received the expected event message.

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".

Wait for Event

The **Wait for event** phase waits for an event from a DCS and allows to automatically synchronize an operation and the DCS or any other operator-run operation.

It can be used for processing requirements, such as:

- **Receive a message from an automation system**
A process has reached a certain stage on the automation system and the MES can continue with this process.
- **Remind the operator of a task**
After the operator has verified that a manufacturing step was completed, the operator must receive a reminder after a defined period of time to proceed with a further procedural step.
This example makes use of the **Time-based trigger** phase, an IPC phase.

Execution

In addition to the instruction text, the **Wait for event** phase displays an information message that shows its status.

- **Waiting for <message identifier> message** indicates that the phase waits to receive a message with the given identifier.
- **<Message identifier> received** indicates that the phase has received the expected message.
- **Wait for <message identifier> aborted** indicates that the operator has recorded an **Abort wait** exception (page 177).

When the phase becomes active, it waits to receive the indicated message. The **Confirm** button is disabled while the phase is waiting. When it receives the message or the operator aborts the wait by recording the corresponding exception, the phase completes automatically.

The phase provides a user-triggered exception to abort the wait.

After completion the phase displays the message identifier and the phase's aborted state in the Execution Window.

The Navigator displays the timestamp when the message was received.



Figure 41: Wait for event during execution



Figure 42: User-triggered exception of Wait for event



Figure 43: Wait for event after phase completion



Figure 44: Wait for event in the Navigator

Phase Design

The characteristics of the **Wait for event** 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 left column provides space for textual instructions.

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

When the phase is completed it shows the same two-column layout.

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).

Event message

Indicates the identifier of the message to be received from the DCS or a **Send event** phase.

It needs to correspond to the identifier defined in the **Event message** process parameter (page 190) of the **Send event** phase and must always be defined. Otherwise the phase cannot complete without exception.

Attribute	Type	Comment
Identifier	String	Identifier of the message to be received from the DCS or the Send event phase. Maximum length is 80 characters.

Abort wait

Represents a user-triggered exception that is accessible from the Exception Window.

The exception allows an operator to abort the wait and thus complete the phase.

It covers incidents when technical reasons prevent the expected message from being sent or received.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception. Since there is no operator interaction for the exception, it is not linked to a 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.

Instruction links

In addition to the permanent process parameters that are always present, the **Wait for event** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.
3. Type an identifier and click the **OK** button.
The system adds the instruction link parameter and the first link definition row to the list of parameters.
Instruction link parameters are generally inserted below all other parameters. Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction link parameter as last parameter.
 - If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
 - If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa. The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.

- Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:



adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.



deletes the currently selected rows.



moves the currently selected row one row up.



moves the currently selected row one row down.



REMOVING INSTRUCTION LINKS

- In the list of parameters, select the instruction link parameter you wish to remove.
- Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction link parameter.

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

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	<p>Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the Instruction link definition process parameter (page 180). Example: Refer to {SOP1270} for guidance.</p> <p>Maximum length is 2000 characters (including HTML tags).</p>

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 179). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

`https://rockwellautomation.com`

`file:///c:/SOP/SOP1492.pdf`

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the Link URL attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction tables

In addition to the permanent process parameters that are always present, the **Wait for event** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
 2. Select the **Instruction Table** type.
The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
 3. Type an identifier and click the **OK** button.
The system adds the parameter definition and the first table row to the list of parameters.
Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction table as last parameter.
 - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
 - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.
- TIPS**

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.
4. Specify the overall appearance of the table:
 - Select the number of columns to define the layout.
 - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
 - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.

- Specify the table rows and their content. The button bar above the rows table provides the following functions:



adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.



deletes the currently selected rows.



moves the currently selected row one row up.



moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.



REMOVING INSTRUCTION TABLES

- In the list of parameters, select the instruction table parameter you wish to remove.
- Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction table with its definition and rows.

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

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: 1 column, 2 columns, 3 columns, 4 columns, 5 columns. Default setting: 1 column.

Attribute	Type	Comment
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. Restriction: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Wait for event** phase provides the following output variables:

Wait aborted

- Data type: Boolean, with the values **true** and **false**
- Usage: The output variable states if the wait state was aborted.
 - The value is `true` if the phase was completed with an **Abort wait** exception.
 - The value is `false` if the phase was completed after having received the expected message.

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".

-
-
- FT PharmaSuite® BB - User Manual DCS Phases
-
-

Send Event

The **Send event** phase sends an event (PharmaSuite message) to a **Wait for event (OES)** phase (page 169) or a **Wait for event** phase (page 175) to support the synchronization between an MES and a DCS or between the operations of an order.

It can be used for processing requirements, such as:

- Send a message to synchronize manufacturing steps
A process has reached a certain stage that allows another waiting manufacturing step to continue with its process.
- Remind the operator of a task
After the operator has verified that a manufacturing step was completed, the operator must receive a reminder after a defined period of time to proceed with a further procedural step.
This example makes use of the **Time-based trigger** phase, an IPC phase.

Execution

The **Send event** phase only displays the instruction text.

When the operator taps the **Confirm** button, the phase sends an event message or a broadcast, depending on its configuration:

- It sends the event message specified with the **Event message** process parameter to either a **Wait for event** phase (page 175) or a **Wait for event (OES)** phase (page 169).
- The sent event aborts the corresponding **Wait for event (OES)** phase, if the **Send abort event** process parameter (page 190) is set to **Yes**.
- The sent event aborts all **Wait for event (OES)** phases of the unit procedure if there is no additional event message specified.
- If configured to run asynchronously with the **Send asynchronous message** process parameter, the sent event is retained for a configurable amount of time. It is delivered to its targeted **Wait for event** phase (page 175) or **Wait for event (OES)** phase (page 169) as soon as the phase becomes active.

TIP

Please note that the asynchronous mode is not available for abort broadcasts that are sent if there is no message specified in the **Event message** process parameter (page 170) of a **Wait for event (OES)** phase and the **Send abort event** process parameter (page 190) is set to **Yes**.

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

- In the **Manual completion** mode, the operator manually confirms the phase thus triggering the event message or broadcast.
- In the **Automatic completion** mode, the phase immediately confirms as soon as it becomes active thus triggering the event message or broadcast without any operator interaction.

After completion the phase displays the message identifier and the timestamp when the message was sent.

The Navigator displays the message identifier and the event type of the message that was sent.



Figure 45: Send event during execution (normal event)



Figure 46: Send event during execution (abort event)



Figure 47: Send event after phase completion (normal event)



Figure 48: Send event after phase completion (abort event)



Figure 49: Send event in the Navigator (normal event)



Figure 50: Send event in the Navigator (abort event)

Phase Design

The characteristics of the **Send event** 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 left column provides space for textual instructions.

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

When the phase is completed it shows the same two-column layout.

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 is automatically completed after the reply to the sent event has been received or the broadcast (abort message) has been sent.

Event message

Indicates the identifier of the message to be sent.

It needs to be defined for all normal event messages.

For abort events, which need to be enabled in the **Send abort event** process parameter (page 190), it is only required if the abort message is targeted at a specific **Wait for event (OES)** phase. If left blank, the message broadcast aborts all **Wait for event (OES)** phases of the unit procedure.

Attribute	Type	Comment
Identifier	String	Identifier of the message to be sent to a Wait for event phase. Maximum length is 80 characters.

Send abort event

Defines if the phase sends an abort event message or a normal message.

Abort messages are only directed at **Wait for event (OES)** messages. By defining a specific identifier in the **Event message** process parameter (page 190), you can control if the abort event applies to a specific phase or to all **Wait for event (OES)** phases of the unit procedure.

Attribute	Type	Comment
Enabled	Boolean	Controls if the phase can send abort events.

Send asynchronous message

Defines if the sent event message is retained until either its targeted **Wait for event** phase (page 175) or **Wait for event (OES)** phase (page 169) becomes active or its **Timeout period** has expired. The asynchronous mode is not available for abort broadcasts that are sent if there is no message specified in the **Event message** process parameter (page 170) of a **Wait for event (OES)** phase and the **Send abort event** process parameter (page 190) is set to **Yes**.

Attribute	Type	Comment
Enabled	Boolean	Controls if the phase runs in the asynchronous mode.
Timeout period	Duration	Defines the retention time of the message in milliseconds.

Skip after failed sending

Represents a system-triggered exception that is displayed in the Exception Window. It provides a way to proceed if the message could not be sent successfully.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception. Since there is no operator interaction for the exception, it is not linked to a 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.

Instruction links

In addition to the permanent process parameters that are always present, the **Send event** phase provides instruction links as optional process parameters, which you can insert if required.

You can add up to ten instruction link parameters.

TIP

Instruction links are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.



ADDING INSTRUCTION LINKS

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Link** type.
The system opens the **Add Instruction Link** dialog to define the instruction link's identifier.

3. Type an identifier and click the **OK** button.

The system adds the instruction link parameter and the first link definition row to the list of parameters.

Instruction link parameters are generally inserted below all other parameters.


Where within the block of instruction link parameters the system adds a new link parameter depends on the current selection in the Parameter Panel:

- If no parameter is selected, the system adds the new instruction link parameter as last parameter.
- If an instruction link parameter is selected, the system adds the new instruction link parameter below the selected one.
- If any other parameter is selected, the system adds the new instruction link parameter as first parameter of the instruction link parameter block.


TIPS


Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa. The identifier of the instruction link parameter is shown as **Identifier** of the link's instruction text parameter.

4. Specify the instruction text to be displayed and mark the link texts by enclosing them in curly brackets.
5. Specify the list of link definitions. Each row of the list defines one hyperlink. The button bar above the list provides the following functions:

 adds a new row to the list. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.

 deletes the currently selected rows.

 moves the currently selected row one row up.

 moves the currently selected row one row down.



REMOVING INSTRUCTION LINKS

1. In the list of parameters, select the instruction link parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction link parameter.

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

Instruction link definition

Defines all links to be available within the instruction text defined with the **Instruction text with links** process parameter (page 193). You can either access a file on the web, such as your intranet, by using the web access syntax or a file on your file system, by using the file access syntax.

Examples:

`https://rockwellautomation.com`

`file:///c:/SOP/SOP1492.pdf`

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the Link URL attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

Instruction text with links

Defines the text of the optional instruction link that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the Instruction link definition process parameter (page 193). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction tables

In addition to the permanent process parameters that are always present, the **Send event** phase provides instruction tables as optional process parameters, which you can insert if required.

You can add up to ten instruction tables with up to 50 table rows.

TIP

Instruction tables are visible in the Phase Preview window in Recipe and Workflow Designer. During execution, however, they are not displayed in the preview mode, but only when the phase becomes active and after its completion.







ADDING INSTRUCTION TABLES

1. Click the **Add parameter** button.
The system opens an option list that holds all optional parameter types available for the phase.
2. Select the **Instruction Table** type.
The system opens the **Add Instruction Table** dialog to define the instruction table's identifier.
3. Type an identifier and click the **OK** button.
The system adds the parameter definition and the first table row to the list of parameters.
Instruction table parameters are generally inserted below all other parameters. Where within the block of instruction table parameters the system adds a new table depends on the current selection in the Parameter Panel:
 - If no parameter is selected, the system adds the new instruction table as last parameter.
 - If an instruction table parameter is selected, the system adds the new instruction table below the selected one.
 - If any other parameter is selected, the system adds the new instruction table as first parameter of the instruction table parameter block.

TIPS

Please note that you can mix the sequence of instruction link and instruction table parameters. This means you can add an instruction link parameter anywhere within a list of instruction table parameters and vice versa.

The identifier of the instruction table is shown as **Identifier** of the table's definition parameter. The identifiers of the individual table rows (**Row-1**, **Row-2**, etc.) are system-defined and not editable.

4. Specify the overall appearance of the table:
 - Select the number of columns to define the layout.
 - Set the width of the first column. If you do not set it to narrow, all columns have equal widths. If you set the first column to narrow, the remaining columns will have equal widths.
 - Define if your table needs to have borders. You can either show all borders of the table and its cells or none.
5. Specify the table rows and their content. The button bar above the rows table provides the following functions:
 -  adds a new row to the table. The row is inserted after the currently selected row. If there is no row selected, the system inserts the new row below the last row.
 -  deletes the currently selected rows.
 -  moves the currently selected row one row up.
 -  moves the currently selected row one row down.

TIP

Please note that the system always retains the consecutive numbering of the rows. If you reorder the table rows or delete rows, the row identifiers are updated accordingly.

**REMOVING INSTRUCTION TABLES**

1. In the list of parameters, select the instruction table parameter you wish to remove.
2. Click the **Remove parameter** button.
The system asks you to confirm the action and then removes the instruction table with its definition and rows.

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

Instruction table definition

Defines the appearance of the optional instruction table that is visible in the preview in Recipe and Workflow Designer, and in the active and completed views during execution.

Attribute	Type	Comment
Table layout	Choice list	Defines the layout of the instruction table holding the instruction texts. Available settings: 1 column, 2 columns, 3 columns, 4 columns, 5 columns . Default setting: 1 column .
First column narrow	Boolean	Defines if the first column of the table shall be narrow.
Show all borders	Boolean	Defines if the borders of the table shall be visible.

Instruction table text

Specifies the instruction texts to be displayed in the individual cells of the instruction table.

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed in a column. Restriction: Maximum length is 2000 characters (including HTML tags).
Column 2	HTML text	
Column 3	HTML text	
Column 4	HTML text	
Column 5	HTML text	

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 operation or the calculated result of several outputs as value into a parameter attribute. When you reference outputs in this manner you need to be aware of the following restrictions:

- Only when a component has been processed does it provide an output that can be fed into another component as attribute value. For this reason, you can never reference an output of a component that is a strict successor of the component 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 component into which you are feeding the output must be able to deal with such an **Undefined** input value.

The **Send event** 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".

A

- Abort wait (parameter)
 - Wait for event • 177
 - Wait for event (OES) • 171
- Add consumption (Show consumed material, parameter) • 147
- Automatic update (Get DCS alarms, parameter) • 112

B

- Batch ID (Create DCS batch, output) • 31
- Boolean (extended) value bundle (Get DCS parameters, output) • 99
- Boolean (extended) value bundle (Get DCS parameters, parameter) • 68
- Boolean value bundle (output)
 - Create DCS batch • 32
 - Get DCS parameters • 98
 - Set DCS parameters • 54
- Boolean value bundle (parameter)
 - Create DCS batch • 24
 - Get DCS parameters • 66
 - Set DCS parameters • 45

C

- Campaign ID (Create DCS batch, output) • 31
- Category (Set DCS parameters, parameter)
 - Boolean value bundle • 46
 - Duration value bundle • 46
 - Numeric value bundle • 47
 - String value bundle • 47
 - Timestamp value bundle • 48
- Completion time (output)
 - Create DCS batch • 35
 - DCA alarm-based trigger • 129
 - Get DCS alarms • 122
 - Get DCS parameters • 106
 - Send event • 198

- Set DCS parameters • 58
- Set order context • 142
- Show consumed material • 155
- Show produced material • 167
- Wait for event • 185
- Wait for event (OES) • 173

- Correct produced material (Show produced material, parameter) • 160
- Create batch manually (Create DCS batch, parameter) • 20
- Create DCS batch • 15
 - Output variable • 31
 - Process parameter • 19

D

- DCS (parameter)
 - Create DCS batch • 19
 - DCS alarm-based trigger • 124
 - Get DCS alarms • 110
 - Get DCS parameters • 63
 - Set DCS parameters • 41
 - Set order context • 134
- DCS alarm-based trigger • 123
 - Output variable • 128
 - Process parameter • 124
- Definition (parameter)
 - Create DCS batch • 20
 - Get DCS parameters • 63
 - Set DCS parameters • 42
- Duration (extended) value bundle (Get DCS parameters, output) • 100
- Duration (extended) value bundle (Get DCS parameters, parameter) • 73
- Duration value bundle (output)
 - Get DCS parameters • 100
 - Set DCS parameters • 55
- Duration value bundle (parameter)
 - Get DCS parameters • 70
 - Set DCS parameters • 46

E

- Event message (parameter)
 - Send event • 190
 - Wait for event • 177
 - Wait for event (OES) • 170
- Expected value configuration (Get DCS parameters, parameter)
 - Boolean (extended) value bundle • 68
 - Boolean value bundle • 66
 - String (extended) value bundle • 85
 - String value bundle • 83
- Expected value definition (Get DCS parameters, parameter)
 - Boolean (extended) value bundle • 69
 - Boolean value bundle • 67
 - String (extended) value bundle • 85
 - String value bundle • 83
- Expected value visible in PEC (Get DCS parameters, parameter)
 - Boolean (extended) value bundle • 69
 - Boolean value bundle • 67
 - String (extended) value bundle • 86
 - String value bundle • 83

F

- Filter criteria (parameter)
 - DCS alarm-based trigger • 124
 - Get DCS alarms • 111

G

- Get DCS alarms • 107
 - Output variable • 120
 - Process parameter • 110
- Get DCS parameters • 59
 - Output variable • 98
 - Process parameter • 63

I

- Identifier (output)
 - Create DCS batch • 35
 - DCS Alarm-based trigger • 129

- Get DCS alarms • 121
- Get DCS parameters • 105
- Send event • 197
- Set DCS parameters • 57
- Set order context • 142
- Show consumed material • 155
- Show produced material • 166
- Wait for event • 184
- Wait for event (OES) • 172
- Instance count (output)
 - Create DCS batch • 35
 - DCS alarm-based trigger • 129
 - Get DCS alarms • 121
 - Get DCS parameters • 105
 - Send event • 197
 - Set DCS parameters • 57
 - Set order context • 142
 - Show consumed material • 155
 - Show produced material • 166
 - Wait for event • 184
 - Wait for event (OES) • 172
- Instruction (parameter)
 - Create DCS batch • 19
 - Get DCS alarms • 110
 - Get DCS parameters • 63
 - Send event • 189
 - Set DCS parameters • 41
 - Set order context • 134
 - Show consumed material • 147
 - Show produced material • 159
 - Wait for event • 176
- Instruction link definition (parameter)
 - Create DCS batch • 27
 - Get DCS alarms • 117
 - Get DCS parameters • 94
 - Send event • 193
 - Set DCS parameters • 50
 - Set order context • 138
 - Show consumed material • 151
 - Show produced material • 162
 - Wait for event • 180

Instruction table definition (parameter)

- Create DCS batch • 30
- Get DCS alarms • 119
- Get DCS parameters • 97
- Send event • 196
- Set DCS parameters • 53
- Set order context • 140
- Show consumed material • 153
- Show produced material • 165
- Wait for event • 182

Instruction table text (parameter)

- Create DCS batch • 30
- Get DCS alarms • 120
- Get DCS parameters • 97
- Send event • 196
- Set DCS parameters • 53
- Set order context • 141
- Show consumed material • 154
- Show produced material • 165
- Wait for event • 183

Instruction text with links (parameter)

- Create DCS batch • 27
- Get DCS alarms • 116
- Get DCS parameters • 94
- Send event • 193
- Set DCS parameters • 50
- Set order context • 137
- Show consumed material • 150
- Show produced material • 162
- Wait for event • 179

Internal batch ID (Create DCS batch, output) • 31

Invalid structure (functional) • 4

L

L-H configuration (Get DCS parameters, parameter)

- Duration (extended) value bundle • 73
- Duration value bundle • 70
- Numeric (extended) value bundle • 80
- Numeric value bundle • 76
- Timestamp (extended) value bundle • 89
- Timestamp value bundle • 87

Limit definition (Get DCS parameters, parameter)

- Duration (extended) value bundle • 75
- Duration value bundle • 72
- Numeric (extended) value bundle • 81
- Numeric value bundle • 78
- Timestamp (extended) value bundle • 91
- Timestamp value bundle • 88

Limits visible in PEC (Get DCS parameters, parameter)

- Duration (extended) value bundle • 75
- Duration value bundle • 72
- Numeric (extended) value bundle • 82
- Numeric value bundle • 78
- Timestamp (extended) value bundle • 91
- Timestamp value bundle • 88

LL-HH configuration (Get DCS parameters, parameter)

- Duration (extended) value bundle • 74
- Duration value bundle • 71
- Numeric (extended) value bundle • 81
- Numeric value bundle • 77
- Timestamp (extended) value bundle • 90
- Timestamp value bundle • 87

M

Master recipe ID (Create DCS batch, output) • 32

Mode (parameter)

- Create DCS batch • 19
- Get DCS parameters • 64
- Send event • 189
- Set DCS parameters • 41
- Set order context • 134

N

Number of converted alarms (Get DCS alarms, output) • 121

Number of retrieved alarms (Get DCS alarms, output) • 121

Numeric (extended) value bundle (Get DCS parameters, output) • 102

Numeric (extended) value bundle (Get DCS parameters, parameter) • 79

Numeric value bundle (output)

Create DCS batch • 33
Get DCS parameters • 101
Set DCS parameters • 55

Numeric value bundle (parameter)

Create DCS batch • 24
Get DCS parameters • 76
Set DCS parameters • 46

O

Output variable

Batch ID (Create DCS batch) • 31
Boolean (extended) value bundle (Get DCS parameters)
• 99
Boolean value (Create DCS batch) • 32
Boolean value bundle (Get DCS parameters) • 98
Boolean value bundle (Set DCS parameters) • 54
Campaign ID (Create DCS batch) • 31
Completion time (Create DCS batch) • 35
Completion time (DCS alarm-based trigger) • 129
Completion time (Get DCS alarms) • 122
Completion time (Get DCS parameters) • 106
Completion time (Send event) • 198
Completion time (Set DCS parameters) • 58
Completion time (Set order context) • 142
Completion time (Show consumed material) • 155
Completion time (Show produced material) • 167
Completion time (Wait for event (OES)) • 173
Completion time (Wait for event) • 185
Create DCS batch • 31
DCS alarm-based trigger • 128
Duration (extended) value bundle (Get DCS
parameters) • 100
Duration value bundle (Get DCS parameters) • 100
Duration value bundle (Set DCS parameters) • 55
Get DCS alarms • 120
Get DCS parameters • 98
Identifier (Create DCS batch) • 35
Identifier (DCS Alarm-based trigger) • 129
Identifier (Get DCS alarms) • 121
Identifier (Get DCS parameters) • 105
Identifier (Send event) • 197

Identifier (Set DCS parameters) • 57
Identifier (Set order context) • 142
Identifier (Show consumed material) • 155
Identifier (Show produced material) • 166
Identifier (Wait for event (OES)) • 172
Identifier (Wait for event) • 184
Instance count (Create DCS batch) • 35
Instance count (DCS alarm-based trigger) • 129
Instance count (Get DCS alarms) • 121
Instance count (Get DCS parameters) • 105
Instance count (Send event) • 197
Instance count (Set DCS parameters) • 57
Instance count (Set order context) • 142
Instance count (Show consumed material) • 155
Instance count (Show produced material) • 166
Instance count (Wait for event (OES)) • 172
Instance count (Wait for event) • 184
Internal batch ID (Create DCS batch) • 31
Master recipe ID (Create DCS batch) • 32
Number of converted alarms (Get DCS alarms) • 121
Number of retrieved alarms (Get DCS alarms) • 121
Numeric (extended) value bundle (Get DCS
parameters) • 102
Numeric value bundle (Create DCS batch) • 33
Numeric value bundle (Get DCS parameters) • 101
Numeric value bundle (Set DCS parameters) • 55
Parameter - Boolean value bundle (Create DCS batch) •
32
Parameter - Boolean value bundle (Set DCS
parameters) • 54
Parameter - Duration value bundle (Set DCS
parameters) • 55
Parameter - Numeric value bundle (Create DCS batch) •
33
Parameter - Numeric value bundle (Set DCS
parameters) • 55
Parameter - String value bundle (Create DCS batch) •
34
Parameter - String value bundle (Set DCS parameters) •
56

- Parameter - Timestamp value bundle (Set DCS parameters) • 57
- Retrieval successful - Boolean (extended) value bundle (Get DCS parameters) • 99
- Retrieval successful - Boolean value bundle (Get DCS parameters) • 99
- Retrieval successful - Duration (extended) value bundle (Get DCS parameters) • 101
- Retrieval successful - Duration value bundle (Get DCS parameters) • 100
- Retrieval successful - Numeric (extended) value bundle (Get DCS parameters) • 102
- Retrieval successful - Numeric value bundle (Get DCS parameters) • 101
- Retrieval successful - String (extended) value bundle (Get DCS parameters) • 103
- Retrieval successful - String value bundle (Get DCS parameters) • 103
- Retrieval successful - Timestamp (extended) value bundle (Get DCS parameters) • 105
- Retrieval successful - Timestamp value bundle (Get DCS parameters) • 104
- Retrieval successful (Get DCS parameters) • 98
- Scale (Create DCS batch) • 32
- Send event • 197
- Set DCS parameters • 54
- Set order context • 141
- Show consumed material • 154
- Show produced material • 166
- Start time (Create DCS batch) • 35
- Start time (DCS alarm-based trigger) • 129
- Start time (Get DCS alarms) • 121
- Start time (Get DCS parameters) • 106
- Start time (Send event) • 198
- Start time (Set DCS parameters) • 57
- Start time (Set order context) • 142
- Start time (Show consumed material) • 155
- Start time (Show produced material) • 167
- Start time (Wait for event (OES)) • 172
- Start time (Wait for event) • 184
- String (extended) value bundle (Get DCS parameters) • 103
- String value bundle (Create DCS batch) • 33
- String value bundle (Get DCS parameters) • 102
- String value bundle (Set DCS parameters) • 56
- Timestamp (extended) value bundle (Get DCS parameters) • 104
- Timestamp value bundle (Get DCS parameters) • 104
- Timestamp value bundle (Set DCS parameters) • 56
- Unit binding bundle (Create DCS batch) • 34
- Unit class - Unit binding bundle (Create DCS batch) • 34
- Unit ID - Unit binding bundle (Create DCS batch) • 34
- Unit of measure - Numeric value bundle (Create DCS batch) • 33
- Value - Boolean (extended) value bundle (Get DCS parameters) • 100
- Value - Boolean value bundle (Create DCS batch) • 32
- Value - Boolean value bundle (Get DCS parameters) • 99
- Value - Boolean value bundle (Set DCS parameters) • 54
- Value - Duration (extended) value bundle (Get DCS parameters) • 101
- Value - Duration value bundle (Get DCS parameters) • 100
- Value - Duration value bundle (Set DCS parameters) • 55
- Value - Numeric (extended) value bundle (Get DCS parameters) • 102
- Value - Numeric value bundle (Create DCS batch) • 33
- Value - Numeric value bundle (Get DCS parameters) • 102
- Value - Numeric value bundle (Set DCS parameters) • 56
- Value - String (extended) value bundle (Get DCS parameters) • 104
- Value - String value bundle (Create DCS batch) • 34
- Value - String value bundle (Get DCS parameters) • 103
- Value - String value bundle (Set DCS parameters) • 56

Value - Timestamp (extended) value bundle (Get DCS parameters) • 105
Value - Timestamp value bundle (Get DCS parameters) • 104
Value - Timestamp value bundle (Set DCS parameters) • 57
Wait aborted (Wait for event (OES)) • 172
Wait aborted (Wait for event) • 184
Wait for event • 183
Wait for event (OES) • 171
Override bundle parameter (Create DCS batch, parameter) • 22
Override bundle parameter (Set DCS parameters, parameter) • 44
Override DCS parameter (Create DCS batch, parameter) • 21
Override DCS parameter value (Get DCS parameters, parameter) • 64
Override definition (Set DCS parameters, parameter) • 44
Override unit binding (Create DCS batch, parameter) • 22

P

Parameter (output)
Boolean value bundle (Create DCS batch) • 32
Boolean value bundle (Set DCS parameters) • 54
Duration value bundle (Set DCS parameters) • 55
Numeric value bundle (Create DCS batch) • 33
Numeric value bundle (Set DCS parameters) • 55
String value bundle (Create DCS batch) • 34
String value bundle (Set DCS parameters) • 56
Timestamp value bundle (Set DCS parameters) • 57
Parameter bundle
Create DCS batch • 23
Get DCS parameters • 65
Set DCS parameters • 45
Parameter type (Get DCS parameters, parameter)
Boolean (extended) value bundle • 69
Boolean value bundle • 67
Duration (extended) value bundle • 75
Duration value bundle • 72
Numeric (extended) value bundle • 82

Numeric value bundle • 79
String (extended) value bundle • 86
String value bundle • 84
Timestamp (extended) value bundle • 92
Timestamp value bundle • 89
Post-ETO alarms exception (DCS alarm-based trigger, parameter) • 128
Process parameter
Abort wait (Wait for event (OES)) • 171
Abort wait (Wait for event) • 177
Add consumption (Show consumed material) • 147
Automatic update (Get DCS alarms) • 112
Boolean (extended) value bundle (Get DCS parameters) • 68
Boolean value bundle (Create DCS batch) • 24
Boolean value bundle (Get DCS parameters) • 66
Boolean value bundle (Set DCS parameters) • 45
Category - Boolean value bundle (Set DCS parameters) • 46
Category - Duration value bundle (Set DCS parameters) • 46
Category - Numeric value bundle (Set DCS parameters) • 47
Category - String value bundle (Set DCS parameters) • 47
Category - Timestamp value bundle (Set DCS parameters) • 48
Correct produced material (Show produced material) • 160
Create batch manually (Create DCS batch) • 20
Create DCS batch • 19
DCS (Create DCS batch) • 19
DCS (DCS alarm-based trigger) • 124
DCS (Get DCS alarms) • 110
DCS (Get DCS parameters) • 63
DCS (Set DCS parameters) • 41
DCS (Set order context) • 134
DCS alarm-based trigger • 124
Definition (Create DCS batch) • 20
Definition (Get DCS parameters) • 63
Definition (Set DCS parameters) • 42

- Duration (extended) value bundle (Get DCS parameters) • 73
- Duration value bundle (Get DCS parameters) • 70
- Duration value bundle (Set DCS parameters) • 46
- Event message (Send event) • 190
- Event message (Wait for event (OES)) • 170
- Event message (Wait for event) • 177
- Expected value configuration - Boolean (extended) value bundle (Get DCS parameters) • 68
- Expected value configuration - Boolean value bundle (Get DCS parameters) • 66
- Expected value configuration - String (extended) value bundle (Get DCS parameters) • 85
- Expected value configuration - String value bundle (Get DCS parameters) • 83
- Expected value definition - Boolean (extended) value bundle (Get DCS parameters) • 69
- Expected value definition - Boolean value bundle (Get DCS parameters) • 67
- Expected value definition - String (extended) value bundle (Get DCS parameters) • 85
- Expected value definition - String value bundle (Get DCS parameters) • 83
- Expected value visible in PEC - Boolean (extended) value bundle (Get DCS parameters) • 69
- Expected value visible in PEC - Boolean value bundle (Get DCS parameters) • 67
- Expected value visible in PEC - String (extended) value bundle (Get DCS parameters) • 86
- Expected value visible in PEC - String value bundle (Get DCS parameters) • 83
- Filter criteria (DCS alarm-based trigger) • 124
- Filter criteria (Get DCS alarms) • 111
- Get DCS alarms • 110
- Get DCS parameters • 63
- Instruction (Create DCS batch) • 19
- Instruction (Get DCS alarms) • 110
- Instruction (Get DCS parameters) • 63
- Instruction (Send event) • 189
- Instruction (Set DCS parameters) • 41
- Instruction (Set order context) • 134
- Instruction (Show consumed material) • 147
- Instruction (Show produced material) • 147
- Instruction (Wait for event) • 176
- Instruction link definition (Create DCS batch) • 27
- Instruction link definition (Get DCS alarms) • 117
- Instruction link definition (Get DCS parameters) • 94
- Instruction link definition (Send event) • 193
- Instruction link definition (Set DCS parameters) • 50
- Instruction link definition (Set order context) • 138
- Instruction link definition (Show consumed material) • 151
- Instruction link definition (Show produced material) • 162
- Instruction link definition (Wait for event) • 180
- Instruction table definition (Create DCS batch) • 30
- Instruction table definition (Get DCS alarms) • 119
- Instruction table definition (Get DCS parameters) • 97
- Instruction table definition (Send event) • 196
- Instruction table definition (Set DCS parameters) • 53
- Instruction table definition (Set order context) • 140
- Instruction table definition (Show consumed material) • 153
- Instruction table definition (Show produced material) • 165
- Instruction table definition (Wait for event) • 182
- Instruction table text (Create DCS batch) • 30
- Instruction table text (Get DCS alarms) • 120
- Instruction table text (Get DCS parameters) • 97
- Instruction table text (Send event) • 196
- Instruction table text (Set DCS parameters) • 53
- Instruction table text (Set order context) • 141
- Instruction table text (Show consumed material) • 154
- Instruction table text (Show produced material) • 165
- Instruction table text (Wait for event) • 183
- Instruction text with links (Create DCS batch) • 27
- Instruction text with links (Get DCS alarms) • 116
- Instruction text with links (Get DCS parameters) • 94
- Instruction text with links (Send event) • 193
- Instruction text with links (Set DCS parameters) • 50
- Instruction text with links (Set order context) • 137

Instruction text with links (Show consumed material) • 150	LL-HH configuration - Duration (extended) value bundle (Get DCS parameters) • 74
Instruction text with links (Show produced material) • 162	LL-HH configuration - Duration value bundle (Get DCS parameters) • 71
Instruction text with links (Wait for event) • 179	LL-HH configuration - Numeric (extended) value bundle (Get DCS parameters) • 81
L-H configuration - Duration (extended) value bundle (Get DCS parameters) • 73	LL-HH configuration - Numeric value bundle (Get DCS parameters) • 77
L-H configuration - Duration value bundle (Get DCS parameters) • 70	LL-HH configuration - Timestamp (extended) value bundle (Get DCS parameters) • 90
L-H configuration - Numeric (extended) value bundle (Get DCS parameters) • 80	LL-HH configuration - Timestamp value bundle (Get DCS parameters) • 87
L-H configuration - Numeric value bundle (Get DCS parameters) • 76	Mode (Create DCS batch) • 19
L-H configuration - Timestamp (extended) value bundle (Get DCS parameters) • 89	Mode (Get DCS parameters) • 64
L-H configuration - Timestamp value bundle (Get DCS parameters) • 87	Mode (Send event) • 189
Limit definition - Duration (extended) value bundle (Get DCS parameters) • 75	Mode (Set DCS parameters) • 41
Limit definition - Duration value bundle (Get DCS parameters) • 72	Mode (Set order context) • 134
Limit definition - Numeric (extended) value bundle (Get DCS parameters) • 81	Numeric (extended) value bundle (Get DCS parameters) • 79
Limit definition - Numeric value bundle (Get DCS parameters) • 78	Numeric value bundle (Create DCS batch) • 24
Limit definition - Timestamp (extended) value bundle (Get DCS parameters) • 91	Numeric value bundle (Get DCS parameters) • 76
Limit definition - Timestamp value bundle (Get DCS parameters) • 88	Numeric value bundle (Set DCS parameters) • 46
Limits visible in PEC - Duration (extended) value bundle (Get DCS parameters) • 75	Override bundle parameter (Create DCS batch) • 22
Limits visible in PEC - Duration value bundle (Get DCS parameters) • 72	Override bundle parameter (Set DCS parameters) • 44
Limits visible in PEC - Numeric (extended) value bundle (Get DCS parameters) • 82	Override DCS parameter (Create DCS batch) • 21
Limits visible in PEC - Numeric value bundle (Get DCS parameters) • 78	Override DCS parameter value (Get DCS parameters) • 64
Limits visible in PEC - Timestamp (extended) value bundle (Get DCS parameters) • 91	Override definition (Set DCS parameters) • 44
Limits visible in PEC - Timestamp value bundle (Get DCS parameters) • 88	Override unit binding (Create DCS batch) • 22
	Parameter type - Boolean (extended) value bundle (Get DCS parameters) • 69
	Parameter type - Boolean value bundle (Get DCS parameters) • 67
	Parameter type - Duration (extended) value bundle (Get DCS parameters) • 75
	Parameter type - Duration value bundle (Get DCS parameters) • 72
	Parameter type - Numeric (extended) value bundle (Get DCS parameters) • 82
	Parameter type - Numeric value bundle (Get DCS parameters) • 79

Parameter type - String (extended) value bundle (Get DCS parameters) • 86

Parameter type - String value bundle (Get DCS parameters) • 84

Parameter type - Timestamp (extended) value bundle (Get DCS parameters) • 92

Parameter type - Timestamp value bundle (Get DCS parameters) • 89

Post-ETO alarms exception (DCS alarm-based trigger) • 128

Record alarms exception (Get DCS alarms) • 114

Remove incorrectly recorded data set (Show consumed material) • 148

Re-send creation request (Create DCS batch) • 23

Re-send order context (Set Order Context) • 135

Re-send request (Set DCS parameters) • 42

Retrieval exception (DCS alarm-based trigger) • 127

Retrieval exception (Get DCS alarms) • 113

Retrieving cycle (DCS alarm-based trigger) • 126

Send abort event (Send event) • 190

Send asynchronous message (Send event) • 190

Send event • 189

Set DCS parameters • 41

Set DCS parameters manually (Set DCS parameters) • 43

Set order context • 134

Set order context manually (Set order context) • 135

Show consumed material • 147

Show produced material • 159

Skip after failed sending (Send event) • 191

String (extended) value bundle (Get DCS parameters) • 84

String value bundle (Create DCS batch) • 25

String value bundle (Get DCS parameters) • 82

String value bundle (Set DCS parameters) • 47

Timeout exception (DCS alarm-based trigger) • 127

Timeout period (DCS alarm-based trigger) • 126

Timestamp (extended) value bundle (Get DCS parameters) • 89

Timestamp value bundle (Get DCS parameters) • 86

Timestamp value bundle (Set DCS parameters) • 47

Unconverted alarms (Get DCS alarms) • 113

Unit binding bundle (Create DCS batch) • 25

Wait for event • 176

Wait for event (OES) • 170

R

Record alarms exception (Get DCS alarms, parameter) • 114

Remove incorrectly recorded data set (Show consumed material, parameter) • 148

Re-send creation request (Create DCS batch, parameter) • 23

Re-send order context (Set order context, parameter) • 135

Re-send request (Set DCS parameters, parameter) • 42

Retrieval exception (parameter)

- DCS alarm-based trigger • 127
- Get DCS alarms • 113

Retrieval successful (output)

- Boolean (extended) value bundle (Get DCS parameters) • 99
- Boolean value bundle (Get DCS parameters) • 99
- Duration (extended) value bundle (Get DCS parameters) • 101
- Duration value bundle (Get DCS parameters) • 99
- Get DCS parameters • 98
- Numeric (extended) value bundle (Get DCS parameters) • 102
- Numeric value bundle (Get DCS parameters) • 101
- String (extended) value bundle (Get DCS parameters) • 103
- String value bundle (Get DCS parameters) • 103
- Timestamp (extended) value bundle (Get DCS parameters) • 105
- Timestamp value bundle (Get DCS parameters) • 104

Retrieving cycle (DCS alarm-based trigger, parameter) • 126

S

Scale (Create DCS batch, output) • 32

Send abort event (Send event, parameter) • 190

Send asynchronous message (send event, parameter) • 190

Send event • 187
 Output variable • 197
 Process parameter • 189
Set DCS parameters • 37
 Output variable • 54
 Process parameter • 41
Set DCS parameters manually (Set DCS parameters, parameter) • 43
Set order context • 131
 Output variable • 141
 Process parameter • 134
Set order context manually (Show consumed material, parameter) • 135
Show consumed material • 145
 Output variable • 154
 Process parameter • 147
Show produced material • 157
 Output variable • 166
 Process parameter • 159
Skip after failed sending (Send event, parameter) • 191
Start time (output)
 Create DCS batch • 35
 DCS alarm-based trigger • 129
 Get DCS alarms • 121
 Get DCS parameters • 106
 Send event • 198
 Set DCS parameters • 57
 Set order context • 142
 Show consumed material • 155
 Show produced material • 167
 Wait for event • 184
String (extended) value bundle (Get DCS parameters, output) • 103
String (extended) value bundle (Get DCS parameters, parameter) • 84
String value bundle (output)
 Create DCS batch • 33
 Get DCS parameters • 102
 Set DCS parameters • 56
String value bundle (parameter)
 Create DCS batch • 25

Get DCS parameters • 82
Set DCS parameters • 47
Structural context
 Materials management with DCS integration • 13
 Synchronization between PharmaSuite and DCS • 13
 Trigger phases • 4
Structure issue (Trigger phases) • 4
 Looped Event-triggered operation after simultaneous branch • 11
 Operations not strictly parallel • 6
 Sequential Event-triggered operations with identical phase reference • 6

T

Timeout exception (DCS alarm-based trigger, parameter) • 127
Timeout period (DCS alarm-based trigger, parameter) • 126
Timestamp (extended) value bundle (Get DCS parameters, output) • 104
Timestamp (extended) value bundle (Get DCS parameters, parameter) • 89
Timestamp value bundle (output)
 Get DCS parameters • 104
 Set DCS parameters • 56
Timestamp value bundle (parameter)
 Get DCS parameters • 86
 Set DCS parameters • 47

U

Unconverted alarms (Get DCS alarms, parameter) • 113
Unit binding bundle (Create DCS batch, output) • 34
Unit binding bundle (Create DCS batch, parameter) • 25
Unit class - Unit binding bundle (Create DCS batch, output) • 34
Unit ID - Unit binding bundle (Create DCS batch, output) • 34
Unit of measure - Numeric value bundle (Create DCS batch, output) • 33

V

Valid structure (functional) • 4

Value (output)

- Boolean (extended) value bundle (Get DCS parameters)
 - 100
- Boolean value bundle (Create DCS batch) • 32
- Boolean value bundle (Get DCS parameters) • 99
- Boolean value bundle (Set DCS parameters) • 54
- Duration (extended) value bundle (Get DCS parameters) • 101
- Duration value bundle (Get DCS parameters) • 100
- Duration value bundle (Set DCS parameters) • 55
- Numeric (extended) value bundle (Get DCS parameters) • 102
- Numeric value bundle (Create DCS batch) • 33
- Numeric value bundle (Get DCS parameters) • 102
- Numeric value bundle (Set DCS parameters) • 56
- String (extended) value bundle (Get DCS parameters) • 104
- String value bundle (Create DCS batch) • 34
- String value bundle (Get DCS parameters) • 103
- String value bundle (Set DCS parameters) • 56
- Timestamp (extended) value bundle (Get DCS parameters) • 105
- Timestamp value bundle (Get DCS parameters) • 104
- Timestamp value bundle (Set DCS parameters) • 57

W

- Wait aborted (Wait for event, output) • 184
- Wait for event • 175
 - Output variable • 183
 - Process parameter • 176
- Wait for event (OES) • 169
 - Output variable • 171
 - Process parameter • 170