LISTEN.
THINK.
SOLVE.®

PharmaSuite®

DCS PHASES

RELEASE 8.4

FUNCTIONAL REQUIREMENT SPECIFICATION

PUBLICATION PSFDRSC-RM001D-EN-E-DECEMBER-2017
Supersedes publication PSFDDC-RM001C-EN-E

Allen-Bradley • Rockwell Software
Rockwell Automation

Contact Rockwell See contact information provided in your maintenance contract.

Copyright Notice

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

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

FactoryTalk Shop Operations Server, FactoryTalk ProductionCentre, FactoryTalk Administration Console, FactoryTalk Automation Platform, and FactoryTalk Security.
Operational Data Store, ODS, Plant Operations, Process Designer, Shop Operations, Rockwell Software CPGSuite, and Rockwell Software AutoSuite.

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

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

ControlNet is a registered trademark of ControlNet International.

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

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

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

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

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

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

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

Contents

Figures

Introduction

This document details the requirements of the functions implemented by the phases specific to communicating with a Distributed Control System (DCS). The phases are executed in the Production Execution Client of PharmaSuite.
In order to use the DCS phases, the following prerequisites apply:

- Rockwell Automation's DCS Adapter has to be installed and configured
  For details, please refer to "Functional Requirement Specification DCS Adapter" [A2] (page 95) and "Technical Manual DCS Adapter" [A3] (page 95).
- A Distributed Control System has to be set up and configured. The alarms processed by the DCS phase are completely based on the alarms provided by the DCS.

Each requirement is composed of a name and a unique identifier (e.g. Instruction (SR0500.8.1)). If a requirement's meaning is for requirement grouping only, the identifier is appended by a plus sign (e.g. Process parameters (SR0500.8+)).
For requirements with **Framework capability** as identifier, see "Functional Requirement Specification Execution Framework" for their unique identifier, [A1] (page 95).

The revision history (page 97) lists the changes made to the document with PharmaSuite 8.3 as the comparison baseline. Changes related to a requirement are marked as "Editorial", "Update", "New", or "Deleted", changes to the additional context information are marked as "Context information-related".

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

- window and dialog titles
- menu functions
- panel, tab, and button names
- box labels
- object properties and their values (e.g. status).

`Monospaced typeface`	Designates code examples.

Automation Integration with DCS Phases

DCS Alarm-related Phases and Operations

The DCS alarm-related phases support different scenarios:

- Collecting alarms from the DCS at the end of the batch run
  To collect and annotate the alarms from the DCS, use the **Get DCS alarms** phase in a normal operation which is modeled after the DCS batch run is finished.
- Collecting alarms in parallel to normal processing
  To collect and annotate the alarms from the DCS as soon as they occur, we recommend to use event-triggered operations with the **DCS alarm-based trigger** phase.
  PharmaSuite for Production Execution uses event-triggered operations (ETOs) as templates to create specific runs (ETO instances), which then are executed by the operator.
  The creation of the runs is triggered either manually by an operator or automatically by a trigger phase. In the context of a Distributed Control System (DCS), this is done by the **DCS alarm-based trigger** phase.

  The typical structure of a recipe with an event-triggered operation can be modeled with the following characteristics:

  - An operation (**DCS Run**) with the **Event-triggered** capability represents an ETO.
    The runs can be created automatically by a trigger phase (**DCS alarm-based trigger**) if the operation also holds the **Trigger-enabled** capability.
    This operation contains the **Get DCS alarms** phase with the user interface to collect the alarms.
  - For automatic triggers, the trigger phases are located in an operation (**DCS Triggers**) that holds the **Server-run** capability. Thus the operation and its phases are not visible in the Production Execution Client.
  - Both the **DCS Run** and the **DCS Triggers** operations are located on parallel branches, which means both operations become active during execution at the same time.
  - The **Trigger-enabled** capability allows to reference specific trigger phases (**DCS alarm-based trigger**) that typically run on a server (within a server-run operation).

One ETO can reference multiple trigger phases.
One trigger phase can be referenced from multiple ETOs.

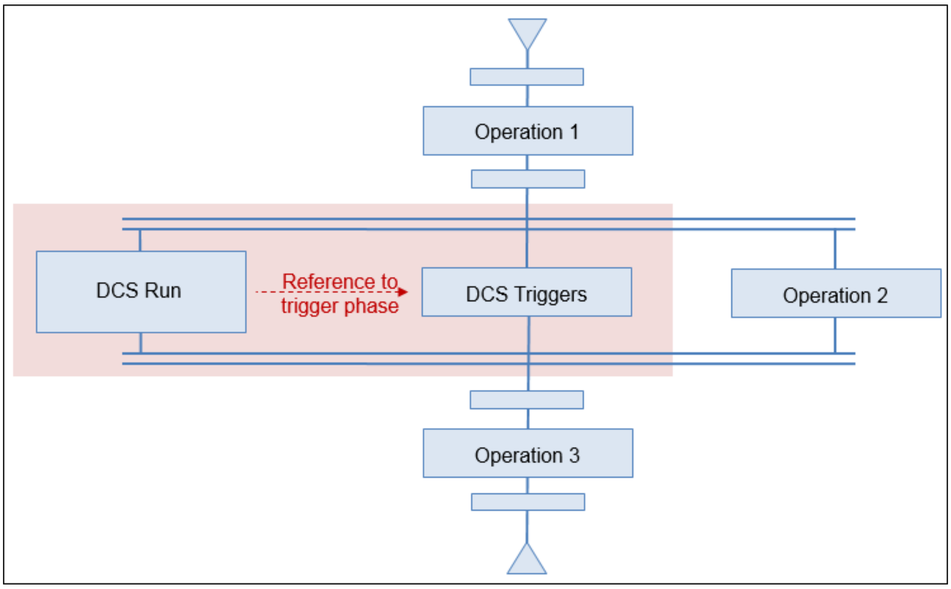


Figure 1: DCS-related phases and event-triggered operation within a recipe

Trigger Phases

The following trigger phases are available:

- DCS alarm-based trigger (page 83)
  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.
  The phase is designed for being run on a server without user interaction.

The following rules apply with respect to start and completion of trigger processing of a trigger phase. For details, see **Business Logic (SR0510.2+)** of the DCS alarm-based trigger phase (page 83).

- A trigger phase becomes active automatically according to SFC, but trigger processing does not start until at least one related ETO template has become active. That means that the template is visible in the Cockpit of all running Production Execution Clients, according to their station-level dispatching.
- If none of the related ETO templates becomes active, the trigger phase is completed automatically after its timeout period has elapsed.
- If trigger processing has started due to active ETO templates, the trigger phase is completed automatically as soon as there is no related ETO template active any more. That means that the active ETO templates have been removed from the Cockpit by the operator.

- In case a unit procedure is paused by the operator, also the trigger processing of the trigger phases is paused. When the pause period of the unit procedure is ended by the operator, the trigger processing continues based on a new re-calculated trigger schedule.

Chapter 3

Create DCS Batch Phase (SR0520+)

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

An example use case is:

- 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.

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

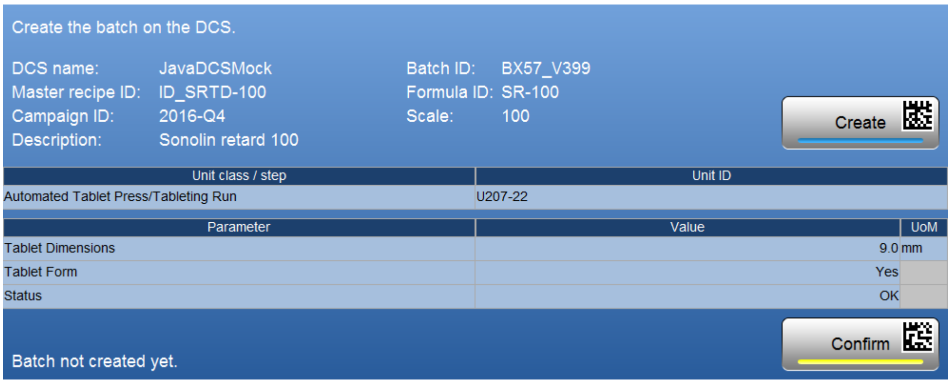
- In the **Manual completion** mode, the operator manually triggers the creation of the batch.
- In the **Automatic completion** mode, the phase creates the batch and is completed automatically without any operator interaction.

The DCS, batch, master recipe, and further product-specific parameters are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 11).

Anomalies that occur during processing are covered by the phase exception handling (page 20) (e.g. re-send creation request).

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
Master recipe ID: ID_SRTD-100
Campaign ID: 2016-Q4
Description: Sonolin retard 100

Batch ID: BX57_V399
Formula ID: SR-100
Scale: 100

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.

Figure 2: Create DCS batch during execution

Layout

The phase provides individual layouts for its representation during execution (page 8), in the Navigator (page 11), and in the sub-report (page 11).

Representation during Execution (SR0520.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0520.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0520.8.1)** process parameter (page 15))
2. DCS name: <logical name>
   Batch ID: <identifier>
   Master recipe ID: <identifier>
   Formula ID: <identifier>
   Campaign ID: <identifier>
   Scale: <value>
   Description: <text>
   (taken from **DCS (SR0520.8.3)** process parameter (page 15) and **Definition (SR0520.8.4)** process parameter (page 15))
3. List of up to 50 bundle parameters in the order of the bundle-specific process parameters:
   - **Unit Binding Bundle**:
     List of unit binding parameters
     (taken from the unit binding-specific **Master bundle identifier (SR0520.8.10)** process parameter (page 19))
   - **String Value Bundle**:
     List of string value parameters
     (taken from the string value-specific **Master bundle identifier (SR0520.8.11)** process parameter (page 19))
   - **Boolean Value Bundle**:
     List of boolean value parameters
     (taken from the boolean value-specific **Master bundle identifier (SR0520.8.12)** process parameter (page 20))
   - **Numeric Value Bundle**:
     List of numeric value parameters
     (taken from the numeric value-specific **Master bundle identifier (SR0520.8.13)** process parameter (page 20))
4. **Create** button (disabled).
5. **Confirm** button (disabled).

Active mode (SR0520.1.2)

1. <Instruction text>
   (taken from **Instruction (SR0520.8.1)** process parameter (page 15))

2. DCS name: <logical name>
   Batch ID: <identifier>
   Master recipe ID: <identifier>
   Formula ID: <identifier>
   Campaign ID: <identifier>
   Scale: <value>
   Description: <text>
   (taken from **DCS (SR0520.8.3)** process parameter (page 15) and **Definition (SR0520.8.4)** process parameter (page 15))

3. List of up to 50 bundle parameters in the order of the bundle-specific process parameters:

   - **Unit Binding Bundle**:
     List of unit binding parameters
     (taken from the unit binding-specific **Master bundle identifier (SR0520.8.10)** process parameter (page 19))

   - **String Value Bundle**:
     List of string value parameters
     (taken from the string value-specific **Master bundle identifier (SR0520.8.11)** process parameter (page 19))

   - **Boolean Value Bundle**:
     List of boolean value parameters
     (taken from the boolean value-specific **Master bundle identifier (SR0520.8.12)** process parameter (page 20))

   - **Numeric Value Bundle**:
     List of numeric value parameters
     (taken from the numeric value-specific **Master bundle identifier (SR0520.8.13)** process parameter (page 20))

4. Batch creation-specific information:

   - Batch not created yet.
     Initial message.

   - Batch created (<timestamp>): <batch ID>
     Batch has been created with the **Create** button or the **Re-send creation request (SR0520.3.1.2)** user-triggered exception (page 21).

- Batch created manually (<timestamp>): <batch ID>
  Batch has been created with the **Create batch manually (SR0520.3.1.1)** user-triggered exception (page 21). If no batch identifier was specified, <batch ID> is DEFAULT_BATCH_ID.
- Batch creation failed.
  Batch could not be created with the **Create** button.
- Batch re-creation failed.
  Batch could not be created with **Re-send creation request (SR0520.3.1.2)** user-triggered exception (page 21).

5. **Create** button (disabled after a batch creation request has been sent).
6. **Confirm** button.

Completed mode (SR0520.1.3)

1. <Instruction text>
   (taken from **Instruction (SR0520.8.1)** process parameter (page 15))
2. DCS name: <logical name>
   Batch ID: <identifier>
   Master recipe ID: <identifier>
   Formula ID: <identifier>
   Campaign ID: <identifier>
   Scale: <value>
   Description: <text>
   (taken from **DCS (SR0520.8.3)** process parameter (page 15) and **Definition (SR0520.8.4)** process parameter (page 15))
3. List of up to 50 bundle parameters in the order of the bundle-specific process parameters:
   - **Unit Binding Bundle**:
     List of unit binding parameters
     (taken from the unit binding-specific **Master bundle identifier (SR0520.8.10)** process parameter (page 19))
   - **String Value Bundle**:
     List of string value parameters
     (taken from the string value-specific **Master bundle identifier (SR0520.8.11)** process parameter (page 19))
   - **Boolean Value Bundle**:
     List of boolean value parameters
     (taken from the boolean value-specific **Master bundle identifier (SR0520.8.12)** process parameter (page 20))

- **Numeric Value Bundle**:
  List of numeric value parameters
  (taken from the numeric value-specific **Master bundle identifier (SR0520.8.13)** process parameter (page 20))

4. Batch creation-specific information:
   - Batch created (<timestamp>): <batch ID>
   - Batch created manually (<timestamp>): <batch ID>
5. **Create** button (disabled).
6. **Confirm** button (completed).

Representation in Navigator (SR0520.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>
  - Example:
    Create BS110 batch

Information column (SR0520.4.1)

- <Batch identifier>
  - Example: BX2859_V15900

Action column

- There are no actions available.

Representation in Sub-report (SR0520.5+)

The sub-report contains the following information:

Common sub-report elements (Framework capability)

- <Start time>
- <Completion time>
- <Unit procedure> / <operation> / <phase>
- <Work center> / <station> / <device> - <phase completion user>

Sub-report elements (SR0520.5.1)

- Instruction text
- DCS name: <logical name>
  Batch ID: <identifier>
  Master recipe ID: <identifier>
  Formula ID: <identifier>
  Campaign ID: <identifier>
  Scale: <value>
  Description: <string>
- Table of unit binding bundle process parameters (in the order of the process parameters).
  - Unit class/step
    Unit ID
- Table of data type-specific bundle process parameters (in the order of the process parameters).
  - Parameter
    Value
    UoM (only for numeric values)
- Table of created batches (in the order of their creation).
  - Batch ID
    Timestamp

In the grid, the phase displays "N/A" for those entries that cannot be provided due to their context (e.g. a unit of measure for a boolean value) or that have not been defined during recipe or workflow design (e.g. a lower limit for a numeric value).

Business Logic (SR0520.2+)

The phase implements the following business logic.

Phase Mode

Business logic related to phase modes.

Manual completion mode (SR0520.2.1)

- Function: **Manual completion** mode of phase
- Type: Phase mode
- Trigger: Phase becomes active
- Postcondition: Phase is active

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0520.1.2)** layout (page 9).
Operator taps **Create** button	20	Phase creates the batch on the DCS, see **Create batch (SR0520.2.3)** function (page 13).
Phase completion	30	See **Confirm phase (SR0520.2.4)** function (page 14).

Automatic completion mode (SR0520.2.2)

- Function: **Automatic completion** mode of phase
- Type: Phase mode
- Trigger: Phase becomes active
- Postcondition: Phase is active

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0520.1.2)** layout (page 9).
Phase creates batch	20	See **Create batch (SR0520.2.3)** function (page 13). ■ If no error has occurred, continue with the **Confirm phase (SR0520.2.4)** function (page 14). ■ If an error has occurred, phase must be completed manually. See **Manual completion (SR0520.2.1)** mode (page 12).

Main Path

Business logic related to the main path:

Create batch (SR0520.2.3)

- Function: Request batch creation
- Type: Main path
- Trigger: Operator creates batch or **Automatic completion (SR0520.2.2)** mode (page 13) is active
- Postcondition: Phase is active

Step	#	Description
Phase creates batch	10	Phase creates the batch on the DCS and disables the **Create** button.

Step	#	Description
	20.1	If an error has occurred, phase displays the **Batch creation error (SR0520.3.6.1)** error message (page 29). Continue with the **Re-send creation request (SR0520.3.1.2)** user-triggered exception (page 21) or the **Create batch manually (SR0520.3.1.1)** user-triggered exception (page 21). Phase can be completed with the **Confirm phase (SR0520.2.4)** function (page 14).
	20.2	If no error has occurred, continue with the **Confirm phase (SR0520.2.4)** function (page 14).

Confirm phase (SR0520.2.4)

- Function: Completion of phase
- Type: Main path
- Trigger: Operator confirms phase or **Automatic completion (SR0520.2.2)** mode (page 13) is active
- Postcondition: Phase is completed

Step	#	Description
In **Manual completion (SR0520.2.1)** mode (page 12): Operator confirms phase	10	Operator confirms the creation of the batch.
Phase performs completion checks	20	If no batch has been created, phase displays the **No batch created (SR0520.3.6.2)** error message (page 29). The phase cannot be completed. Continue with the **Create batch manually (SR0520.3.1.1)** user-triggered exception (page 21).
Phase completion	30	Phase is completed.

Process Parameters (SR0520.8+)

The following process parameters define the behavior of the phase.

BASIC PARAMETERS

Instruction (SR0520.8.1)

- For recent changes, see revision history (page 97).

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

Mode (SR0520.8.2)

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 (SR0520.8.3)

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.

Definition (SR0520.8.4)

Attribute	Type	Comment
Batch ID	String	Defines the identifier of the batch to be created on the DCS. This attribute is required by the DCS Adapter.
Master recipe ID	String	Defines the identifier of the master recipe to be used. This attribute is required by the DCS Adapter.

Attribute	Type	Comment
Formula ID	String	Defines the identifier of the formula to be used.
Description	String	Defines the description of the batch to be created.
Campaign ID	String	Defines the identifier of the campaign to be used.
Scale	BigDecimal	Defines the scale of the batch in percent.

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Create batch manually (SR0520.8.5)

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.

See also **Create batch manually (SR0520.3.1.1)** user-triggered exception (page 21).

Override DCS parameter (SR0520.8.6)

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.

See also **Override DCS parameter (SR0520.3.1.3)** user-triggered exception (page 22).

Override bundle parameter (SR0520.8.7)

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.

See also **Override bundle parameter (String value) (SR0520.3.1.5)** user-triggered exception (page 25), **Override bundle parameter (Boolean value) (SR0520.3.1.6)** user-triggered exception (page 26), and **Override bundle parameter (Numeric value) (SR0520.3.1.7)** user-triggered exception (page 27).

Override unit binding (SR0520.8.8)

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.

See also **Override unit binding (SR0520.3.1.4)** user-triggered exception (page 24).

Re-send creation request (SR0520.8.9)

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.

See also **Re-send creation request (SR0520.3.1.2)** user-triggered exception (page 21).

Unit Binding Bundle

Bundle process parameters (Framework capability)

For the master process parameter of a bundle, its internal identifier is populated from the bundle identifier.

For all other process parameters of the bundle, their internal identifier is a concatenation of the bundle identifier and the process parameter name.

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Master (Bundle identifier) (SR0520.8.10)

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

String Value Bundle

Bundle process parameters (Framework capability)

For the master process parameter of a bundle, its internal identifier is populated from the bundle identifier.

For all other process parameters of the bundle, their internal identifier is a concatenation of the bundle identifier and the process parameter name.

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Master (Bundle identifier) (SR0520.8.11)

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be used.
Value	String	Defines the value of the parameter.

Boolean Value Bundle

Bundle process parameters (Framework capability)

For the master process parameter of a bundle, its internal identifier is populated from the bundle identifier.

For all other process parameters of the bundle, their internal identifier is a concatenation of the bundle identifier and the process parameter name.

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Master (Bundle identifier) (SR0520.8.12)

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be used.
Value	Boolean	Defines the value of the parameter.

Numeric Value Bundle

Bundle process parameters (Framework capability)

For the master process parameter of a bundle, its internal identifier is populated from the bundle identifier.

For all other process parameters of the bundle, their internal identifier is a concatenation of the bundle identifier and the process parameter name.

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Master (Bundle identifier) (SR0520.8.13)

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

Exceptions (SR0520.3+)

The phase supports user-defined, user-triggered (page 21), system-triggered (page 20), and post-completion exceptions (page 28) and their configuration by means of process parameters (page 14).

User-defined exceptions cannot be configured by process parameters since they are provided by the framework and independent of phases.

System-triggered Exceptions

There are no system-triggered exceptions available.

User-triggered Exceptions (SR0520.3.1+)

A user-triggered exception is represented in the list of available user-triggered exceptions in the Exception Window, as the description of the exception, and in the batch report.

The following user-triggered exceptions are available.

Create batch manually (SR0520.3.1.1)

The **Create batch manually** exception allows an operator to create the DCS batch manually based on the internal batch identifier of the DCS batch. If the identifier is unknown, the phase uses DEFAULT_BATCH_ID as batch identifier to create the DCS batch on the DCS.
With the exception, the phase overrides the recently created batch, if available.

Representation of the exception:

- Instruction:
  Create the DCS batch manually.
  Batch ID <value>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Create batch manually (SR0520.8.5)** process parameter (page 16))
  Manual entry: < batch identifier>
  If no batch identifier was specified:
  Manual entry: N/A (DEFAULT_BATCH_ID)
  - Example:
    Batch created manually
    Manual entry: BS101338_04Jul2016_2111_PV51400

Create batch manually - Logic (SR0520.3.1.1.1)

- Trigger: Exception is selected
- Postcondition: DCS batch is created on the DCS

Step	#	Description
Operator confirms and signs exception	10	Phase records exception, updates batch creation-specific information in the **Active mode (SR0520.1.2)** layout (page 9), and disables the **Create** button in the **Active mode (SR0520.1.2)** layout (page 9)

Re-send creation request (SR0520.3.1.2)

The **Re-send creation request** exception allows an operator to re-send the DCS batch creation request to the DCS.
With the exception, the phase overrides the recently created batch, if available.

Representation of the exception:

- Instruction:
  Re-send the DCS batch creation request.
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Re-send creation request (SR0520.8.9)** process parameter (page 18))
  Batch ID: <value>
  - Example:
    Batch creation repeated.
    Batch ID: BX2859_V15900

Re-send creation request - Logic (SR0520.3.1.2.1)

- Trigger: Exception is selected
- Postcondition: DCS batch is created on the DCS

Step	#	Description
Operator confirms and signs exception	10	Phase records the exception and sends a creation request to the DCS.
Phase receives feedback from the DCS	20	Phase adds a comment with the batch creation result ■ Batch created (<timestamp>): <batch ID> ■ Batch re-creation failed to the exception, updates batch creation-specific information in the **Active mode (SR0520.1.2)** layout (page 9), and disables the **Create** button in the **Active mode (SR0520.1.2)** layout (page 9). If the batch re-creation fails, the phase cannot be completed (see **Confirm phase (SR0520.2.4)** function (page 14)).

Override DCS parameter (SR0520.3.1.3)

The **Override DCS parameter** exception allows an operator to override the DCS parameters defined in the master recipe/workflow.

Representation of the exception:

- Instruction:
  Override a defined DCS parameter.
  Batch ID:
  Current value <value>
  Override value <value>

Master recipe ID:
Current value <value>
Override value <value>
Formula ID:
Current value <value>
Override value <value>
Campaign ID:
Current value <value>
Override value <value>
Scale:
Current value <value>
Override value <value>
Description:
Current value <value>
Override value <value>
Confirm button.

- Exception text:
  <Exception text>
  (taken from **Override DCS parameter (SR0520.8.6)** process parameter (page 16))
  Batch ID:
  Old value: <old value>
  New value: <new value>
  Master recipe ID:
  Old value: <old value>
  New value: <new value>

  - Example:
    Parameter overridden.
    Batch ID:
    Old value: BX2859_V15901
    New value: BX2859_V15900
    Master recipe ID:
    Old value: ID_SRTD_002
    New value: ID_SRTD_001

Override DCS parameter - Logic (SR0520.3.1.3.1)

- Trigger: Exception is selected
- Postcondition: Value of DCS parameter is overridden

Step	#	Description
Operator confirms and signs exception	10	Phase records exception.
Operator interaction	20	Operator taps **Create** button in the **Active mode (SR0520.1.2)** layout (page 9) or triggers the **Re-send creation request (SR0520.3.1.2)** user-triggered exception (page 21) to create a DCS batch with the new values.

UNIT BINDING BUNDLE

Override unit binding (SR0520.3.1.4)

The **Override unit binding** exception allows an operator to override the unit identifiers of the unit binding-specific process parameter defined in the master recipe/workflow.

There is one exception per unit binding-specific process parameter.

Representation during exception handling:

- Instruction:
  Override defined unit binding:
  <Unit class/step>
  (unit binding-specific **Master (bundle identifier) (SR0520.8.10)** process parameter (page 19))
  Current unit ID <value>
  New unit ID <value>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override unit binding (SR0520.8.8)** process parameter (page 17))
  <Unit class/step>
  (unit binding-specific **Master (bundle identifier) (SR0520.8.10)** process parameter (page 19))
  Old unit ID: <value>
  New unit ID: <value>
  - Example:
    Unit binding overridden.
    Step 1
    Old unit ID: U207-22-067
    New unit ID: U207-22-068

Override unit binding - Logic (SR0520.3.1.4.1)

- Trigger: Exception is selected
- Postcondition: Value of unit binding parameter is overridden

Step	#	Description
Operator confirms and signs exception	10	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0520.1.2)** layout (page 9).
Operator interaction	20	Operator taps **Create** button in the **Active mode (SR0520.1.2)** layout (page 9) or triggers the **Re-send creation request (SR0520.3.1.2)** user-triggered exception (page 21) to create a DCS batch with the new values.

STRING VALUE BUNDLE

Override bundle parameter (SR0520.3.1.5)

The **Override bundle parameter** exception allows an operator to override the values of a string value-specific process parameter defined in the master recipe/workflow.

There is one exception per string value-specific process parameter.

Representation during exception handling:

- Instruction:
  <Parameter>
  (taken from the numeric-specific **Master (bundle identifier) (SR0520.8.13)** process parameter (page 20))
  Override defined value:
  Current value <value>
  Override value <value>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override bundle parameter (SR0520.8.7)** process parameter (page 17))
  <Parameter>
  (taken from the numeric-specific **Master (bundle identifier) (SR0520.8.13)** process parameter (page 20))
  Old value: <value>
  New value: <value>
  - Example:
    Parameter overridden.
    Target batch ID
    Old value: BX40060
    New value: BX40061

Override bundle parameter - Logic (SR0520.3.1.5.1)

- Trigger: Exception is selected
- Postcondition: Value of bundle parameter is overridden

Step	#	Description
Operator confirms and signs exception	10	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0520.1.2)** layout (page 9).
Operator interaction	20	Operator taps **Create** button in the **Active mode (SR0520.1.2)** layout (page 9) or triggers the **Re-send creation request (SR0520.3.1.2)** user-triggered exception (page 21) to create a DCS batch with the new values.

BOOLEAN VALUE BUNDLE

Override bundle parameter (SR0520.3.1.6)

The **Override bundle parameter** exception allows an operator to override the values of a boolean value-specific process parameter defined in the master recipe/workflow.

There is one exception per boolean value-specific process parameter.

Representation during exception handling:

- Instruction:
  <Parameter>
  (taken from the boolean-specific **Master (bundle identifier) (SR0520.8.12)** process parameter (page 20))
  Override defined value:
  Current value <value>
  Override value <value>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override bundle parameter (SR0520.8.7)** process parameter (page 17))
  <Parameter>
  (taken from the boolean-specific **Master (bundle identifier) (SR0520.8.12)** process parameter (page 20))
  Old value: <value>
  New value: <value>
  - Example:
    Parameter overridden.
    Pressure indicator active

Old value: No
New value: Yes

Override bundle parameter - Logic (SR0520.3.1.6.1)

- Trigger: Exception is selected
- Postcondition: Value of bundle parameter is overridden

Step	#	Description
Operator confirms and signs exception	10	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0520.1.2)** layout (page 9).
Operator interaction	20	Operator taps **Create** button in the **Active mode (SR0520.1.2)** layout (page 9) or triggers the **Re-send creation request (SR0520.3.1.2)** user-triggered exception (page 21) to create a DCS batch with the new values.

Numeric Value Bundle

Override bundle parameter (SR0520.3.1.7)

The **Override bundle parameter** exception allows an operator to override the values of a numeric value-specific process parameter defined in the master recipe/workflow.

There is one exception per numeric value-specific process parameter.

Representation during exception handling:

- Instruction:
  <Parameter>
  (taken from the numeric-specific **Master (bundle identifier) (SR0520.8.13)** process parameter (page 20))
  Override defined value:
  Current value <value> <unit of measure>
  Override value <value> <unit of measure>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override bundle parameter (SR0520.8.7)** process parameter (page 17))
  <Parameter>
  (taken from the numeric-specific **Master (bundle identifier) (SR0520.8.13)** process parameter (page 20))
  Old value: <value> <unit of measure>
  New value: <value> <unit of measure>

- Example:
  Parameter overridden.
  Table size
  Old value: 7.8 mm
  New value: 7.9 mm

Override bundle parameter - Logic (SR0520.3.1.7.1)

- Trigger: Exception is selected
- Postcondition: Value of bundle parameter is overridden

Step	#	Description
Operator confirms and signs exception	10	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0520.1.2)** layout (page 9).
Operator interaction	20	Operator taps **Create** button in the **Active mode (SR0520.1.2)** layout (page 9) or triggers the **Re-send creation request (SR0520.3.1.2)** user-triggered exception (page 21) to create a DCS batch with the new values.

Post-completion Exceptions

There are no post-completion exceptions available.

Information Messages

There are no information messages available.

Questions

There are no questions available.

Decisions

There are no decisions available.

Error Messages (SR0520.3.6+)

Error messages are represented in an error message dialog containing a message type-specific icon, the error message, and an **OK** button.

The following error messages are available to inform the operator about error conditions.

Batch creation error (SR0520.3.6.1)

UI text	Comment
Cannot create the batch on the DCS.	Message pack: PhaseCreateDCSBatch<version> Message ID: BatchCreateError_message

The **Details** button provides access to more specific technical information.

No batch created (SR0520.3.6.2)

UI text	Comment
You have to create a batch before you can confirm.	Message pack: PhaseCreateDCSBatch<version> Message ID: NoBatchCreated_ErrorMsg

Output Variables (SR0520.9+)

The following output variables are available to reference the phase's output.

Instance count (Framework capability)

- Data type: Long
- 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 (Framework capability)

- Data type: Timestamp
- Usage: The output variable provides the start time of the phase.

Completion time (Framework capability)

- Data type: Timestamp
- Usage: The output variable provides the completion time of the phase.

Identifier (Framework capability)

- Data type: String
- Usage: The output variable provides the identifier of the phase.

Batch ID (SR0520.9.1)

- Data type: String
- Usage: The output variable provides the identifier of the DCS batch that was created on the DCS.

Internal batch ID (SR0520.9.2)

- Data type: String
- Usage: The output variable provides the identifier of the internal batch that depends on the used DCS.

Master recipe ID (SR0520.9.3)

- Data type: String
- Usage: The output variable provides the identifier of the master recipe that was used to create the DCS batch.

Formula ID (SR0520.9.4)

- Data type: String
- Usage: The output variable provides the identifier of the formula that was used to create the DCS batch.

Campaign ID (SR0520.9.5)

- Data type: String
- Usage: The output variable provides the identifier of the campaign that was used to create the DCS batch.

Scale (SR0520.9.6)

- Data type: BigDecimal
- Usage: The output variable provides the scale that was used to create the DCS batch.

Unit Binding Bundle

Bundle output variable (Framework capability)

For all output variables of the same bundle, the output variable identifier is a concatenation of the bundle identifier and the output variable name.

This framework capability refers to **Bundle Output Variable (SR3146.9.7.4.2)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Unit class (SR0520.9.7)

- Data type: String
- Usage: The output variable provides the unit class that was used to create the DCS batch.

Unit ID (SR0520.9.8)

- Data type: String
- Usage: The output variable provides the identifier of the unit that was used to create the DCS batch.

String Value Bundle

Bundle output variable (Framework capability)

For all output variables of the same bundle, the output variable identifier is a concatenation of the bundle identifier and the output variable name.

This framework capability refers to **Bundle Output Variable (SR3146.9.7.4.2)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Parameter (SR0520.9.9)

- Data type: String
- Usage: The output variable provides the identifier of the parameter that was used to create the DCS batch.

Value (SR0520.9.10)

- Data type: String
- Usage: The output variable provides the value of the parameter that was used to create the DCS batch.

Boolean Value Bundle

Bundle output variable (Framework capability)

For all output variables of the same bundle, the output variable identifier is a concatenation of the bundle identifier and the output variable name.

This framework capability refers to **Bundle Output Variable (SR3146.9.7.4.2)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Parameter (SR0520.9.11)

- Data type: String
- Usage: The output variable provides the identifier of the parameter that was used to create the DCS batch.

Value (SR0520.9.12)

- Data type: Boolean
- Usage: The output variable provides the value of the parameter that was used to create the DCS batch.

Numeric Value Bundle

Bundle output variable (Framework capability)

For all output variables of the same bundle, the output variable identifier is a concatenation of the bundle identifier and the output variable name.

This framework capability refers to **Bundle Output Variable (SR3146.9.7.4.2)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Parameter (SR0520.9.13)

- Data type: String
- Usage: The output variable provides the identifier of the parameter that was used to create the DCS batch.

Value (SR0520.9.14)

- Data type: BigDecimal
- Usage: The output variable provides the value of the parameter that was used to create the DCS batch.

Unit of measure (SR0520.9.15)

- Data type: String
- Usage: The output variable provides the unit of measure of the parameter that was used to create the DCS batch.

Configuration Keys (SR0520.11+)

The following configuration keys are available to configure the phase's behavior.

Message broker URL (SR0520.11.1)

- **Phase/DCS/DCSMessageBrokerURL**
- **Type**: String
- **Value**: tcp://<hostname>:<port>
- **Description**: Specifies the URL of the messaging server used for DCS communication. Replace <hostname> by the name of the host to be used and <port> by the port number of the host to be used.
  If no value is set, the URL defined in the **MessageBrokerURL** configuration key is used.
- **Evaluated**: When a DCS phase communicates with a DCS.
- **Range**: N/A

Messaging timeout (SR0520.11.2)

- **Phase/DCS/DCSMessagingTimeout**
- **Type**: Long
- **Value**: 2
- **Description**: The messaging timeout for DCS communication in seconds.
  If no value is set, the timeout defined in the **MessagingTimeout** configuration key is used (converted into seconds).
- **Evaluated**: When a DCS phase communicates with a DCS.
- **Range**: N/A

Get DCS Batch Values Phase (SR0525+)

The **Get DCS batch values** phase allows an operator to retrieve values from a batch processed on a DCS.

An example use case is:

- Adding data to the batch report from a batch executed on an automation system
  The phase can retrieve specific values of a batch executed on a DCS. The values are added to the phase-specific sub-report and thus available in the batch report.

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

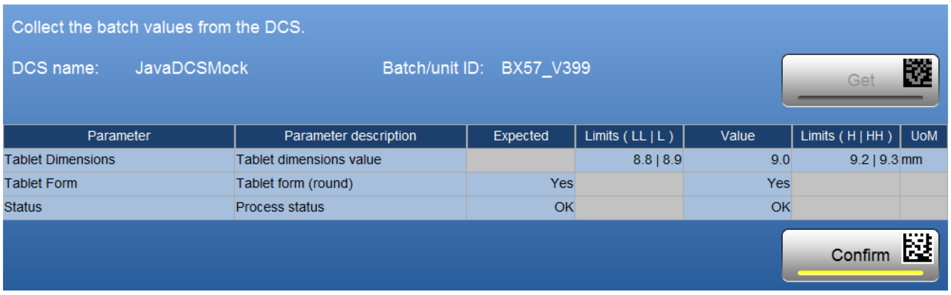
- In the **Manual completion** mode, the operator manually triggers retrieving the batch values.
- In the **Automatic completion** mode, the phase retrieves the batch values and is completed automatically without any operator interaction.

The DCS, batch, the batch parameters, and their values are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 40).

Anomalies that occur during processing are covered by the phase exception handling (page 52) (e.g. limit violation).

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 batch values from the DCS.

DCS name: JavaDCSMock

Batch/unit ID: BX57_V399

Parameter	Parameter description	Expected	Limits (LL | L)	Value	Limits (H | HH)	UoM
Tablet Dimensions	Tablet dimensions value		8.8 | 8.9	9.0	9.2 | 9.3	mm
Tablet Form	Tablet form (round)	Yes		Yes		
Status	Process status	OK		OK		

Figure 3: Get DCS batch values during execution

Layout

The phase provides individual layouts for its representation during execution (page 36), in the Navigator (page 39), and in the sub-report (page 40).

Representation during Execution (SR0525.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0525.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0525.8.1)** process parameter (page 45))
2. DCS name: <logical name>
   Batch/unit ID: <identifier>
   (taken from **DCS (SR0525.8.3)** process parameter (page 45) and **Definition (SR0525.8.4)** process parameter (page 45))
3. List of up to 50 bundle parameters in the order of the bundle-specific process parameters:
   - **String Value Bundle:**
     List of string value parameters
     - Parameter
       Parameter description
       (taken from the string value-specific **Master bundle identifier (SR0525.8.6)** process parameter (page 46))
     - Expected
       (taken from the string value-specific **Expected value definition (SR0525.8.8)** process parameter (page 47))
     - Value
   - **Boolean Value Bundle:**
     List of boolean value parameters
     - Parameter
       Parameter description
       (taken from the boolean value-specific **Master bundle identifier (SR0525.8.9)** process parameter (page 48))
     - Expected
       (taken from the boolean value-specific **Expected value definition (SR0525.8.11)** process parameter (page 49))
     - Value

- **Numeric Value Bundle:**
  List of numeric value parameters
  - Parameter
    Parameter description
    UoM
    (taken from the numeric value-specific **Master (bundle identifier) (SR0525.8.12)** process parameter (page 49))
  - Limits (LL | L)
    Limits (H | HH)
    (taken from the numeric value-specific **Limit definition (SR0525.8.15)** process parameter (page 51))
  - Value

4. **Get** button (disabled).
5. **Confirm** button (disabled).

Active mode (SR0525.1.2)

1. <Instruction text>
   (taken from **Instruction (SR0525.8.1)** process parameter (page 45))
2. DCS name: <logical name>
   Batch/unit ID: <identifier>
   (taken from **DCS (SR0525.8.3)** process parameter (page 45) and **Definition (SR0525.8.4)** process parameter (page 45))
3. List of up to 50 bundle parameters in the order of the bundle-specific process parameters:
   - **String Value Bundle**:
     List of string value parameters
     - Parameter
       Parameter description
       (taken from the string value-specific **Master bundle identifier (SR0525.8.6)** process parameter (page 46))
     - Expected
       (taken from the string value-specific **Expected value definition (SR0525.8.8)** process parameter (page 47))
     - Value
       For the representation of the value, see **Get batch values (SR0525.2.3)** function (page 42).

- **Boolean Value Bundle**:
  List of boolean value parameters
  - Parameter
    Parameter description
    (taken from the boolean value-specific **Master bundle identifier (SR0525.8.9)** process parameter (page 48))
  - Expected
    (taken from the boolean value-specific **Expected value definition (SR0525.8.11)** process parameter (page 49))
  - Value
    For the representation of the value, see **Get batch values (SR0525.2.3)** function (page 42).
- **Numeric Value Bundle**:
  List of numeric value parameters
  - Parameter
    Parameter description
    UoM
    (taken from the numeric value-specific **Master (bundle identifier) (SR0525.8.12)** process parameter (page 49))
  - Limits (LL | L)
    Limits (H | HH)
    (taken from the numeric value-specific **Limit definition (SR0525.8.15)** process parameter (page 51))
  - Value
    For the representation of the value, see **Get batch values (SR0525.2.3)** function (page 42).

4. **Get** button.
5. **Confirm** button.

Completed mode (SR0525.1.3)

1. <Instruction text>
   (taken from **Instruction (SR0525.8.1)** process parameter (page 45))
2. DCS name: <logical name>
   Batch/unit ID: <identifier>
   (taken from **DCS (SR0525.8.3)** process parameter (page 45) and **Definition (SR0525.8.4)** process parameter (page 45))
3. List of up to 50 bundle parameters in the order of the bundle-specific process parameters:

- **String Value Bundle**:
  List of string value parameters
  - Parameter
    Parameter description
    (taken from the string value-specific **Master bundle identifier (SR0525.8.6)** process parameter (page 46))
  - Expected
    (taken from the string value-specific **Expected value definition (SR0525.8.8)** process parameter (page 47))
  - Value
- **Boolean Value Bundle**:
  List of boolean value parameters
  - Parameter
    Parameter description
    (taken from the boolean value-specific **Master bundle identifier (SR0525.8.9)** process parameter (page 48))
  - Expected
    (taken from the boolean value-specific **Expected value definition (SR0525.8.11)** process parameter (page 49))
  - Value
- **Numeric Value Bundle**:
  List of numeric value parameters
  - Parameter
    Parameter description
    UoM
    (taken from the numeric value-specific **Master (bundle identifier) (SR0525.8.12)** process parameter (page 49))
  - Limits (LL | L)
    Limits (H | HH)
    (taken from the numeric value-specific **Limit definition (SR0525.8.15)** process parameter (page 51))
  - Value

4. **Get** button (disabled).
5. **Confirm** button (completed).

Representation in Navigator (SR0525.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>
  - Example:
    Retrieve values from BS110 batch

Information column (SR0525.4.1)

- <Batch identifier>
  - Example: BX2859_V15900

Action column

- There are no actions available.

Representation in Sub-report (SR0525.5+)

The sub-report contains the following information:

Common sub-report elements (Framework capability)

- <Start time>
- <Completion time>
- <Unit procedure> / <operation> / <phase>
- <Work center> / <station> / <device> **-** <phase completion user>

Sub-report elements (SR0525.5.1)

- Instruction text
- DCS name: <logical name>
  Batch/unit ID: <identifier>
- Table of values that have been retrieved during execution (in the order of the process parameters).
  - List of numeric values
    - Parameter
    - Parameter description
    - Limits (LL | L)
    - Value
    - Limits (H | HH)
    - UoM

- List of string values
    - Parameter
    - Parameter description
    - Expected value
    - Value
- List of boolean values
    - Parameter
    - Parameter description
    - Expected value

In the grid, the phase displays "N/A" for those entries that cannot be provided due to their context (e.g. a unit of measure for a boolean value) or that have not been defined during recipe or workflow design (e.g. a lower limit for a numeric value).

Business Logic (SR0525.2+)

The phase implements the following business logic.

Phase Mode

Business logic related to phase modes.

Manual completion mode (SR0525.2.1)

- Function: **Manual completion** mode of phase
- Type: Phase mode
- Trigger: Phase becomes active
- Postcondition: Phase is active

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0525.1.2)** layout (page 37).
Operator taps **Get** button	20	The phase reads the batch values from the DCS batch, see **Get batch values (SR0525.2.3)** function (page 42). Each time the **Get** button is used, all of the batch values are read unless they have already been read or overridden.
Phase completion	30	See **Confirm phase (SR0525.2.4)** function (page 44).

Automatic completion mode (SR0525.2.2)

- Function: **Automatic completion** mode of phase
- Type: Phase mode
- Trigger: Phase becomes active
- Postcondition: Phase is active

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0525.1.2)** layout (page 37).
Phase gets values	20	See **Get batch values (SR0525.2.3)** function (page 42). ■ If no error has occurred, continue with the **Confirm phase (SR0525.2.4)** function (page 44). ■ If an error or warning has occurred, phase must be completed manually. See **Manual completion (SR0525.2.1)** mode (page 41).

Main Path

Business logic related to the main path:

Get batch values (SR0525.2.3)

- Function: Retrieve batch values
- Type: Main path
- Trigger: Operator retrieves batch values or **Automatic completion (SR0525.2.2)** mode (page 42) is active
- Postcondition: Phase is active

Step	#	Description
Phase checks manual override	10	If a value has been overridden with the **Override DCS batch value (Numeric value) (SR0525.3.1.3)** user-triggered exception (page 57), **Override DCS batch value (String value) (SR0525.3.1.1)** user-triggered exception (page 54), or **Override DCS batch value (Boolean value) (SR0525.3.1.2)** user-triggered exception (page 56) and the exception has been signed, the **Get** action cannot be executed for such a value; phase displays **Override value recorded (SR0525.3.4.1)** information message (page 60).
Phase checks for read batch values	15	If a batch value has already been read from the DCS batch, the **Get** action is not executed for the value.

Step	#	Description
Phase gets values	20	Phase reads the remaining batch values and disables the **Get** button as soon as there are no batch values that have not yet been read or overridden. The order of the **Numeric value - Master (bundle identifier) (SR0525.8.12)** process parameters (page 49), **String value - Master (bundle identifier) (SR0525.8.6)** process parameters (page 46), and **Boolean value - Master (bundle identifier) (SR0525.8.9)** process parameters (page 48) defines the read sequence of the batch values. If any issue occurs, phase behavior is as follows: ■ Phase does not display a value, ■ changes cell background to red, ■ appends "(X)" to the "empty value", and ■ displays **No get result error (SR0525.3.6.1)** error message (page 61).
Validation	20.1	➢ **Numeric Value Bundle** Phase checks the numeric value against the settings of the **Limit definition (SR0525.8.15)** process parameter (page 51). Limits are checked in the following order: LL/HH » L/H. If the check is violated, phase creates the **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52) for a numeric value. ➢ **String Value Bundle** Phase checks the string value against the settings of the **Expected value definition (SR0525.8.8)** process parameter (page 47). If the check is violated, phase creates the **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52) for a string value. ➢ **Boolean Value Bundle** Phase checks the boolean value against the settings of the **Expected value definition (SR0525.8.11)** process parameter (page 49). If the check is violated, phase creates the **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52) for a boolean value. If a check is violated, phase changes cell background to yellow. After the exception has been signed, phase changes cell background to the default and adds the exception marker to the value's cell. If no check is violated, phase returns to the **Active mode (SR0525.1.2)** layout (page 37).

Step	#	Description
	30	If applicable, continue with the **Override DCS batch value (Numeric value) (SR0525.3.1.3)** user-triggered exception (page 57), **Override DCS batch value (String value) (SR0525.3.1.1)** user-triggered exception (page 54), or **Override DCS batch value (Boolean value) (SR0525.3.1.2)** user-triggered exception (page 56). Phase can be completed with the **Confirm phase (SR0525.2.4)** function (page 44).

Confirm phase (SR0525.2.4)

- Function: Completion of phase
- Type: Main path
- Trigger: Operator confirms phase or **Automatic completion (SR0525.2.2)** mode (page 42) is active
- Postcondition: Phase is completed

Step	#	Description
In **Manual completion (SR0525.2.1)** mode (page 41): Operator confirms phase	10	Operator confirms the batch values.
Phase performs completion checks	20	If one of the following issues occurs, phase cannot be completed: - In **Manual completion (SR0525.2.1)** mode (page 41), the **Get** button has not been used. - Not all batch values that were defined have been read. Phase displays **Recorded values incomplete (SR0525.3.6.2)** error message (page 61). If a validation check fails, phase creates the **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52). Continue with the **Override DCS batch value (Numeric value) (SR0525.3.1.3)** user-triggered exception (page 57), **Override DCS batch value (String value) (SR0525.3.1.1)** user-triggered exception (page 54), or **Override DCS batch value (Boolean value) (SR0525.3.1.2)** user-triggered exception (page 56).
Phase completion	30	Phase is completed.

Process Parameters (SR0525.8+)

The following process parameters define the behavior of the phase.

BASIC PARAMETERS

Instruction (SR0525.8.1)

➢ For recent changes, see revision history (page 97).

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

Mode (SR0525.8.2)

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.

DCS (SR0525.8.3)

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.

Definition (SR0525.8.4)

Attribute	Type	Comment
Batch/unit ID	String	Defines the identifier of the batch/unit to be used for data retrieval.

Override DCS batch value (SR0525.8.5)

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.

See also **Override recorded value (Numeric value) (SR0525.3.1.3)** user-triggered exception (page 57), **Override recorded value (String value) (SR0525.3.1.1)** user-triggered exception (page 54), and **Override recorded value (Boolean value) (SR0525.3.1.2)** user-triggered exception (page 56).

String Value Bundle

Bundle process parameters (Framework capability)

For the master process parameter of a bundle, its internal identifier is populated from the bundle identifier.

For all other process parameters of the bundle, their internal identifier is a concatenation of the bundle identifier and the process parameter name.

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Master (Bundle identifier) (SR0525.8.6)

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read.
Parameter description	String	Defines an alias for the parameter.
Parameter path	String	Defines the path of the parameter.

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Expected value configuration (SR0525.8.7)

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 47) 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.

See also **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52).

Expected value definition (SR0525.8.8)

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

See also **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52).

Boolean Value Bundle

Bundle process parameters (Framework capability)

For the master process parameter of a bundle, its internal identifier is populated from the bundle identifier.

For all other process parameters of the bundle, their internal identifier is a concatenation of the bundle identifier and the process parameter name.

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Master (Bundle identifier) (SR0525.8.9)

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read.
Parameter description	String	Defines an alias for the parameter.
Parameter path	String	Defines the path of the parameter.

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Expected value configuration (SR0525.8.10)

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 49) 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.

See also **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52).

Expected value definition (SR0525.8.11)

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

See also **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52).

Numeric Value Bundle

Bundle process parameters (Framework capability)

For the master process parameter of a bundle, its internal identifier is populated from the bundle identifier.

For all other process parameters of the bundle, their internal identifier is a concatenation of the bundle identifier and the process parameter name.

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Master (Bundle identifier) (SR0525.8.12)

Attribute	Type	Comment
Parameter	String	Defines the identifier of the parameter to be read.
Parameter description	String	Defines an alias for the parameter.
Parameter path	String	Defines the path of the parameter.
UoM	Unit of measure	Must match a unit of measure available within PharmaSuite.

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

L-H configuration (SR0525.8.13)

If the checks are activated for the available limit ranges, the checks are performed in the following order:

1. LL-HH (defined with the **LL-HH configuration (SR0525.8.14)** process parameter (page 50))
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 51) 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.

See also **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52).

LL-HH configuration (SR0525.8.14)

If the checks are activated for the available limit ranges, the checks are performed in the following order:

1. LL-HH
2. L-H (defined with the **L-H configuration (SR0525.8.13)** process parameter (page 49))

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 51) 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.

See also **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52).

Limit definition (SR0525.8.15)

The following rule applies to the attributes:

- LL limit < L limit < H limit < HH limit

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

See also **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52).

Exceptions (SR0525.3+)

The phase supports user-defined, user-triggered (page 54), system-triggered (page 52), and post-completion exceptions (page 59) and their configuration by means of process parameters (page 45).

User-defined exceptions cannot be configured by process parameters since they are provided by the framework and independent of phases.

System-triggered Exceptions (SR0525.3.2+)

➢ For recent changes, see revision history (page 97).

A system-triggered exception is represented in a message dialog along with an **Exception** button, in the Exception Window as the read-only description of the exception, and in the batch report.

The following system-triggered exceptions are available.

Limit violation (SR0525.3.2.1)

If several checks fail during the execution of the **Get** action, the exceptions are combined and displayed in a single exception. The highest risk assessment of all related exceptions and its related signature privilege apply.

Representation of the exception:

- A violation of limits or expected values has occurred.
- List of up to 50 bundle parameters in the order of the bundle-specific process parameters:
  - **String Value Bundle**:
    Exception text:
    <Exception text>
    (taken from **Expected value configuration (SR0525.8.7)** process parameter (page 47))
    Parameter: <identifier> (<parameter description>)
    (taken from the string-specific **Master (bundle identifier) (SR0525.8.6)** process parameter (page 46))
    Expected value: <expected value>
    (taken from **Expected value definition (SR0525.8.8)** process parameter (page 47))
    Actual value: <batch value>
    - Example:
      Expected value violation confirmed.
      Parameter: VisualCheckResult (VisualCheckResult-10)
      Expected value: Dark blue
      Actual value: Light blue

- **Boolean Value Bundle**:
  Exception text:
  <Exception text>
  (taken from **Expected value configuration (SR0525.8.10)** process parameter (page 48))
  Parameter: <identifier> (<parameter description>)
  (taken from the boolean-specific **Master (bundle identifier) (SR0525.8.9)** process parameter (page 48))
  Expected value <expected value>
  (taken from **Expected value definition (SR0525.8.11)** process parameter (page 49))
  Actual value: <batch value>

  - Example:
    Expected value violation confirmed.
    Parameter: HeatingPerformed (HeatingPerformed-10)
    Expected value: Yes
    Actual value: No

- **Numeric Value Bundle**:
  Exception text:
  <Exception text>
  (taken from **L-H configuration (SR0525.8.13)** process parameter (page 49) or **LL-HH configuration (SR0525.8.14)** process parameter (page 50))
  Parameter: <identifier> (<parameter description>)
  (taken from the numeric-specific **Master (bundle identifier) (SR0525.8.12)** process parameter (page 49))
  <Affected limit, L, LL, H, HH>: <limit value>
  (taken from **Limit definition (SR0525.8.15)** process parameter (page 51))
  Actual value: <batch value>

  - Example:
    Limit violation confirmed.
    Parameter: AgitatorSpeed (AgitatorSpeed-10)
    LL limit: 300 rpm
    Actual value: 200 rpm

Limit violation - Logic (SR0525.3.2.1.1)

- Trigger: Check has failed
- Postcondition: Exception is recorded

Step	#	Description
Operator accepts exceptional situation	10	Phase shows exception description to be signed.

Step	#	Description
Operator signs exception	20	Phase records exception.

Multiple system-triggered exceptions (SR0525.3.2.2)

In case multiple system-triggered exceptions occur, only one combined exception (system-triggered exception) is recorded including information about all exceptions. The highest risk assessment of all related exceptions and its related signature privilege apply.

User-triggered Exceptions (SR0525.3.1+)

A user-triggered exception is represented in the list of available user-triggered exceptions in the Exception Window, as the description of the exception, and in the batch report.

The following user-triggered exceptions are available.

String Value Bundle

Override DCS batch value (SR0525.3.1.1)

The **Override DCS batch value** exception allows an operator to override the string value read from the DCS batch.

There is one exception per string value-specific process parameter.
Parameters for which an error has been detected are displayed at the top of the list properties.

Representation during exception handling:

- Instruction:
  <Parameter> (<parameter description>)
  (taken from the string-specific **Master (bundle identifier) (SR0525.8.6)** process parameter (page 46))
  Override recorded value:
  Current value: <current value>
  Override value: <value>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override DCS batch value (SR0525.8.5)** process parameter (page 46))
  <Parameter> (<parameter description>)
  (taken from the string-specific **Master (bundle identifier) (SR0525.8.6)** process parameter (page 46))
  Old value: <old value>
  New value: <new value>

- Example:
  Value overridden.
  Target batch ID (Target batch ID-10)
  Old value: BX40060
  New value: BX40060

Override DCS batch value - Logic (SR0525.3.1.1.1)

➢ For recent changes, see revision history (page 97).

- Trigger: Exception is selected
- Postcondition: String value is set

Step	**#**	**Description**
Operator triggers exception	10	Phase displays Exception Window.
	20	Operator enters value.
Operator confirms exception	30	If the related check is enabled, phase checks the string value against the settings of the **Expected value definition (SR0525.8.8)** process parameter (page 47).
	30.1	If the expected value is violated, phase displays a corresponding message dialog with an **Exception** button, the exception text (taken from **Expected value configuration (SR0525.8.7)** process parameter (page 47)), the parameter identifier, the affected limit, and the actual value.
Operator accepts exceptional situation	30.1.1	Phase displays only one combined exception (user-triggered exception), including both exception texts from the override-value-exception and from the violation of the expected value (see **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52) for a string value).
Operator cancels exceptional situation	30.1.2	Phase requires the operator to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).
	30.2	If the expected value is not violated or no check applies, the override value-related exception is displayed.
	30.3	If the following issue occurs, phase displays an error message: ■ Override value is missing, **No value overridden (SR0525.3.6.5)** error message (page 62). Phase shows exception description to be signed according to **Override DCS batch value (SR0525.8.5)** process parameter (page 46).
Operator signs exception	40	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0525.1.2)** layout (page 37).

BOOLEAN VALUE BUNDLE

Override DCS batch value (SR0525.3.1.2)

The **Override DCS batch value** exception allows an operator to override the boolean value read from the DCS batch.

There is one exception per boolean value-specific process parameter.
Parameters for which an error has been detected are displayed at the top of the list properties.

Representation during exception handling:

- Instruction:
  <Parameter> (<parameter description>)
  (taken from the boolean-specific **Master (bundle identifier) (SR0525.8.9)** process parameter (page 48))
  Override recorded value:
  Current value: <current value>
  Override value: <value>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override DCS batch value (SR0525.8.5)** process parameter (page 46))
  <Parameter> (<parameter description>)
  (taken from the boolean-specific **Master (bundle identifier) (SR0525.8.9)** process parameter (page 48))
  Old value: <old value>
  New value: <new value>
  - Example:
    Value overridden.
    Pressure indicator active (Pressure indicator active-10)
    Old value: No
    New value: Yes

Override DCS batch value - Logic (SR0525.3.1.2.1)

➢ For recent changes, see revision history (page 97).

- Trigger: Exception is selected
- Postcondition: Boolean value is set

Step	#	Description
Operator triggers exception	10	Phase displays Exception Window.

Step	#	Description
	20	Operator selects value.
Operator confirms exception	30	If the check is enabled, phase checks the boolean value against the settings of the **Expected value definition (SR0525.8.11)** process parameter (page 49).
	30.1	If the expected value is violated, phase displays a corresponding message dialog with an **Exception** button, the exception text (taken from **Expected value configuration (SR0525.8.10)** process parameter (page 48)), the parameter identifier, the affected limit, and the actual value.
Operator accepts exceptional situation	30.1.1	Phase displays only one combined exception (user-triggered exception), including both exception texts from the override-value-exception and from the violation of the expected value (see **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52) for a boolean value).
Operator cancels exceptional situation	30.1.2	Phase requires the operator to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).
	30.2	If the expected value is not violated or no check applies, the override value-related exception is displayed.
	30.3	If the following issue occurs, phase displays an error message: ■ Override value is missing, **No value overridden (SR0525.3.6.6)** error message (page 62). Phase shows exception description to be signed according to **Override DCS batch value (SR0525.8.5)** process parameter (page 46).
Operator signs exception	40	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0525.1.2)** layout (page 37).

NUMERIC VALUE BUNDLE

Override DCS batch value (SR0525.3.1.3)

The **Override DCS batch value** exception allows an operator to override the numeric value read from the DCS batch.

There is one exception per numeric value-specific process parameter.
Parameters for which an error has been detected are displayed at the top of the list properties.

Representation during exception handling:

- Instruction:
  <Parameter> (<parameter description>)
  (taken from the numeric-specific **Master (bundle identifier) (SR0525.8.12)** process parameter (page 49))
  Override recorded value:
  Current value: <current value> <UoM>
  Override value: <new value> <UoM>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override DCS batch value (SR0525.8.5)** process parameter (page 46))
  <Parameter> (<parameter description>)
  (taken from the numeric-specific **Master (bundle identifier) (SR0525.8.12)** process parameter (page 49))
  Old value: <value> <UoM>
  New value: <value> <UoM>
  - Example:
    Value overridden.
    Table size (Table size-10)
    Old value: 7.8 mm
    New value: 7.9 mm

Override DCS batch value - Logic (SR0525.3.1.3.1)

➢ For recent changes, see revision history (page 97).

- Trigger: Exception is selected
- Postcondition: Numeric value is set

Step	#	Description
Operator triggers exception	10	Phase displays Exception Window.
	20	Operator enters values.
Operator confirms exception	30	If the following issue occurs, phase displays an error message: ■ Data format does not match, **Invalid data format error (SR0525.3.6.3)** error message (page 62). If the related check is enabled, phase checks the numeric value against the settings of the **Limit definition (SR0525.8.15)** process parameter (page 51). Limits are checked in the following order: LL/HH » L/H.

Step	#	Description
	30.1	If a limit is violated, phase displays a corresponding message dialog with an **Exception** button, the exception text (taken from **L-H configuration (SR0525.8.13)** process parameter (page 49) or **LL-HH configuration (SR0525.8.14)** process parameter (page 50)), the parameter identifier, the affected limit, and the actual value.
Operator accepts exceptional situation	30.1.1	Phase displays only one combined exception (user-triggered exception), including both exception texts from the override-value-exception and from the limit violation (see **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52) for a numeric value).
Operator cancels exceptional situation	30.1.2	Phase requires the operator to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).
	30.2	If no limit is violated or no check applies, the override value-related exception is displayed.
	30.3	If the following issue occurs, phase displays an error message: ■ Override value is missing, **No value overridden (SR0525.3.6.4)** error message (page 62). Phase shows exception description to be signed according to **Override DCS batch value (SR0525.8.5)** process parameter (page 46).
Operator signs exception	40	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0525.1.2)** layout (page 37).

NOT BUNDLE-SPECIFIC

Multiple exceptions (SR0525.3.1.4)

In case an **Override DCS batch value (Numeric value) (SR0525.3.1.3)** user-triggered exception (page 57), **Override DCS batch value (String value) (SR0525.3.1.1)** user-triggered exception (page 54), or **Override DCS batch value (Boolean value) (SR0525.3.1.2)** user-triggered exception (page 56) coincides with the **Limit violation (SR0525.3.2.1)** system-triggered exception (page 52), only one combined exception (user-triggered exception) is recorded including information about all related exceptions. The highest risk assessment of all related exceptions and its related signature privilege apply.

Post-completion Exceptions

There are no post-completion exceptions available.

Information Messages (SR0525.3.4+)

Information messages are represented in an information dialog containing a message type-specific icon, the information message, and an **OK** button.

The following information messages are available to inform the operator about how to proceed.

Override value recorded (SR0525.3.4.1)

UI text	Comment
1. Values have already been overridden manually. 2. <list of parameters>	1. Message pack: PhaseDCSGetDCSBatchValues<version> Message ID: OverriddenValuesList_Info 2. Message pack: PhaseDCSGetDCSBatchValues<version> Message ID: Value_Info Potential error cause: **Get** button is used after the **Override recorded value (Numeric value) (SR0525.3.1.3)** user-triggered exception (page 57), **Override recorded value (String value) (SR0525.3.1.1)** user-triggered exception (page 54), or **Override recorded value (Boolean value) (SR0525.3.1.2)** user-triggered exception (page 56) has been signed.

The **Details** button provides access to more specific technical information.

Questions

There are no questions available.

Decisions

There are no decisions available.

Error Messages (SR0525.3.6+)

Error messages are represented in an error message dialog containing a message type-specific icon, the error message, and an **OK** button.

The following error messages are available to inform the operator about error conditions.

Get Batch Value-specific Error Messages

No get result error (SR0525.3.6.1)

UI text	Comment
Configuration error. Please record the values manually.	Message pack: PhaseDCSGetDCSBatchValues<version> Message ID: BatchGetValuesConfigError_message Potential error cause: A required attribute of a process parameter is not configured correctly.
The Get action was not successful. Please record the values manually.	Message pack: PhaseDCSGetDCSBatchValues<version> Message ID: BatchGetValuesError_message Potential error cause: ■ An error occurred while values were retrieved from the DCS. ■ A timeout occurred. ■ Not all values are available on the DCS. ■ Values could not be retrieved (null value was returned).

The **Details** button provides access to more specific technical information.

Phase Completion-specific Error Messages

Recorded values incomplete (SR0525.3.6.2)

UI text	Comment
Cannot confirm. Not all expected values have been recorded. Please record the values manually.	Message pack: PhaseDCSGetDCSBatchValues<version> Message ID: CompletionError_HeaderMsg Potential error cause: The values of the defined batch parameters were not read successfully or overridden. Expected values are still missing.

The **Details** button provides access to more specific technical information.

User-triggered Exception-specific Error Messages

Numeric Value Bundle

Invalid data format error (SR0525.3.6.3)

UI text	Comment
Cannot confirm the overridden value.	Message pack: PhaseDCSGetDCSBatchValues<version> Message ID: OverrideDCSParameterNumberError_Error Potential error cause: The entered text value cannot be converted to a numeric value of the targeted numeric data type.

No value overridden (SR0525.3.6.4)

UI text	Comment
You have to enter an override value before you can confirm.	Message pack: PhaseDCSGetDCSBatchValues<version> Message ID: OverrideNumericValueNotSet_ErrorMsg Potential error cause: No override value was entered before the user-triggered exception was confirmed.

String Value Bundle

No value overridden (SR0525.3.6.5)

UI text	Comment
You have to enter an override value before you can confirm.	Message pack: PhaseDCSGetDCSBatchValues<version> Message ID: OverrideStringValueNotSet_ErrorMsg Potential error cause: No override value was entered before the user-triggered exception was confirmed.

Boolean Value Bundle

No value overridden (SR0525.3.6.6)

UI text	Comment
You have to select a value before you can confirm.	Message pack: PhaseDCSGetDCSBatchValues<version> Message ID: OverrideBooleanValueNotSet_ErrorMsg Potential error cause: No override value was selected before the user-triggered exception was confirmed.

Output Variables (SR0525.9+)

The following output variables are available to reference the phase's output.

Instance count (Framework capability)

- Data type: Long
- 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 (Framework capability)

- Data type: Timestamp
- Usage: The output variable provides the start time of the phase.

Completion time (Framework capability)

- Data type: Timestamp
- Usage: The output variable provides the completion time of the phase.

Identifier (Framework capability)

- Data type: String
- Usage: The output variable provides the identifier of the phase.

Retrieval successful (SR0525.9.1)

- Data type: Boolean
- Values: `true`, `false`
- Usage: The output variable states if the get operation from the DCS batch was successful.
  - The value is `true` if all batch values have been read successfully.
  - The value is `false` if at least one of the batch values could not be read from the DCS batch or has been overridden by using the **Override DCS batch value (Numeric value) (SR0525.3.1.3)** user-triggered exception (page 57), **Override DCS batch value (String value) (SR0525.3.1.1)** user-triggered exception (page 54), or **Override DCS batch value (Boolean value) (SR0525.3.1.2)** user-triggered exception (page 56).

String Value Bundle

Bundle output variable (Framework capability)

For all output variables of the same bundle, the output variable identifier is a concatenation of the bundle identifier and the output variable name.

This framework capability refers to **Bundle Output Variable (SR3146.9.7.4.2)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Value (SR0525.9.2)

- Data type: String
- Usage: The output variable provides the value of the string parameter. The value is Null if N/A is the phase result.

Retrieval successful (SR0525.9.3)

- Data type: Boolean
- Values: `true`, `false`
- Usage: The output variable states if the get operation from the DCS batch was successful.
  - The value is `true` if the batch value of the string parameter has been read successfully.
  - The value is `false` if the batch value of the string parameter could not be read from the DCS batch or has been overridden by using the **Override DCS batch value (SR0525.3.1.2)** user-triggered exception (page 56).

Boolean Value Bundle

Bundle output variable (Framework capability)

For all output variables of the same bundle, the output variable identifier is a concatenation of the bundle identifier and the output variable name.

This framework capability refers to **Bundle Output Variable (SR3146.9.7.4.2)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Value (SR0525.9.4)

- Data type: Boolean
- Usage: The output variable provides the value of the boolean parameter. The value is Null if N/A is the phase result.

Retrieval successful (SR0525.9.5)

- Data type: Boolean
- Values: `true`, `false`
- Usage: The output variable states if the get operation from the DCS batch was successful.
  - The value is `true` if the batch value of the boolean parameter has been read successfully.
  - The value is `false` if the batch value of the boolean parameter could not be read from the DCS batch or has been overridden by using the **Override DCS batch value (SR0525.3.1.2)** user-triggered exception (page 56).

Numeric Value Bundle

Bundle output variable (Framework capability)

For all output variables of the same bundle, the output variable identifier is a concatenation of the bundle identifier and the output variable name.

This framework capability refers to **Bundle Output Variable (SR3146.9.7.4.2)** in "Functional Requirement Specification Recipe and Workflow Management" [A4] (page 95).

Value (SR0525.9.6)

- Data type: BigDecimal
- 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.

Unit of measure (SR0525.9.7)

- Data type: String
- Usage: The output variable provides the unit of measure of the numeric parameter. The value is Null if N/A is the phase result.

Retrieval successful (SR0525.9.8)

- Data type: Boolean
- Values: `true`, `false`
- Usage: The output variable states if the get operation from the DCS batch was successful.
  - The value is `true` if the batch value of the numeric parameter has been read successfully.

- The value is `false` if the batch value of the numeric parameter could not be read from the DCS batch or has been overridden by using the **Override DCS batch value (SR0525.3.1.2)** user-triggered exception (page 56).

Configuration Keys (SR0525.11+)

The following configuration keys are available to configure the phase's behavior.

Message broker URL (SR0525.11.1)

- **Phase/DCS/DCSMessageBrokerURL**
- **Type**: String
- **Value**: tcp://<hostname>:<port>
- **Description**: Specifies the URL of the messaging server used for DCS communication. Replace <hostname> by the name of the host to be used and <port> by the port number of the host to be used.
  If no value is set, the URL defined in the **MessageBrokerURL** configuration key is used.
- **Evaluated**: When a DCS phase communicates with a DCS.
- **Range**: N/A

Messaging timeout (SR0525.11.2)

- **Phase/DCS/DCSMessagingTimeout**
- **Type**: Long
- **Value**: 2
- **Description**: The messaging timeout for DCS communication in seconds.
  If no value is set, the timeout defined in the **MessagingTimeout** configuration key is used (converted into seconds).
- **Evaluated**: When a DCS phase communicates with a DCS.
- **Range**: N/A

Chapter 5

Get DCS Alarms Phase (SR0500+)

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

Example use cases are:

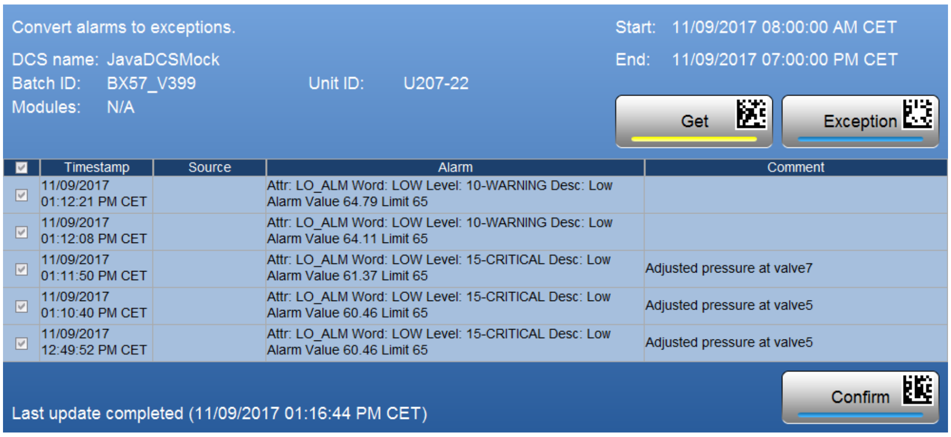
- Concurrent retrieval of alarms for recording in the batch report
  Alarms on the DCS are retrieved every five minutes. In case an alarm has occurred, the alarm can be converted into an exception. Then it is documented in the batch report and included in the review and approval process.
- Retrieval 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.

The batch, unit, alarm timeframe, modules, and the alarms with their details are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 70).

Anomalies that occur during processing are covered by the phase exception handling (page 77) (e.g. unconverted alarms).

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

The Navigator displays the number of retrieved alarms.



Convert alarms to exceptions.

Start: 11/09/2017 08:00:00 AM CET

End: 11/09/2017 07:00:00 PM CET

DCS name: JavaDCSMock

Batch ID: BX57_V399

Unit ID: U207-22

Modules: N/A

☑	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

Last update completed (11/09/2017 01:16:44 PM CET)

Figure 4: Get DCS alarms during execution

Layout

The phase provides individual layouts for its representation during execution (page 68), in the Navigator (page 69), and in the sub-report (page 70).

Representation during Execution (SR0500.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0500.1.1)

➢ For recent changes, see revision history (page 97).

1. <Instruction text>
   (taken from **Instruction (SR0500.8.1)** process parameter (page 74))
2. Start: <timestamp>
   End: <timestamp>
3. DCS name: <logical name>
   Batch ID: <identifier>   Unit ID: <identifier>
   Modules: <list of identifiers>
   (taken from **DCS (SR0500.8.2)** process parameter (page 74) and **Filter criteria (SR0500.8.3)** process parameter (page 74))
4. **Get** button (disabled).
5. **Exception** button (disabled).
6. List of alarms
7. Update-related information:
   - Time of last update unknown.
     No manual or automatic update has been performed yet.
8. **Confirm** button (disabled).

Active mode (SR0500.1.2)

➢ For recent changes, see revision history (page 97).

1. <Instruction text>
   (taken from **Instruction (SR0500.8.1)** process parameter (page 74))
2. Start: <timestamp>
   End: <timestamp>
3. DCS name: <logical name>
   Batch ID: <identifier>   Unit ID: <identifier>
   Modules: <list of identifiers>
   (taken from **DCS (SR0500.8.2)** process parameter (page 74) and **Filter criteria (SR0500.8.3)** process parameter (page 74))

4. **Get** button.
5. **Exception** button (disabled if no alarm is selected).
6. List of alarms
   - Select (checkbox)
   - Timestamp
   - Source
   - Alarm
   - Comment
7. Update-related information:
   - Time of last update unknown.
     No manual or automatic update has been performed yet.
   - Update in progress
     Operator has tapped the **Get** button or the automatic update is currently being performed.
   - Last update completed (<timestamp>)
     After an update has been completed successfully.
   - Last update failed (<timestamp>)
     After an update has failed. Phase displays the **Connection Error (SR0500.3.6.1)** error message (page 80).
8. **Confirm** button.

Completed mode (SR0500.1.3)

1. <Instruction text>
   (taken from **Instruction (SR0500.8.1)** process parameter (page 74))
2. Start: <timestamp>
   End: <timestamp>
3. DCS name: <logical name>
   Batch ID: <identifier> Unit ID: <identifier>
   Modules: <list of identifiers>
   (taken from **DCS (SR0500.8.2)** process parameter (page 74) and **Filter criteria (SR0500.8.3)** process parameter (page 74))
4. Number of retrieved alarms: <value>
5. **Confirm** button (completed).

Representation in Navigator (SR0500.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>
    - Example:
      Collect mixer alarms

Information column (SR0500.4.1)

- <Number of retrieved alarms>
    - Example: 27

Action column

- There are no actions available.

Representation in Sub-report (SR0500.5+)

The sub-report contains the following information:

Common sub-report elements (Framework capability)

- <Start time>
- <Completion time>
- <Unit procedure> / <operation> / <phase>
- <Work center> / <station> / <device> **-** <phase completion user>

Sub-report elements (SR0500.5.1)

- DCS name: <logical name>
  Batch ID: <identifier>
  Unit ID: <identifier>
  Start: <timestamp>
  End: <timestamp>
  Modules: <list of identifiers>
- List of alarms
    - Timestamp
    - Source
    - Alarm
    - Comment
    - Alarm converted

Business Logic (SR0500.2+)

The phase implements the following business logic.

Retrieve alarms without automatic update (SR0500.2.1)

- Function: Manually retrieve alarms from the DCS
- Trigger: Phase becomes active
- Postcondition: Phase is completed

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0500.1.2)** layout (page 68) with an empty list of alarms.
Operator taps **Get** button	20	Phase retrieves alarms from the DCS. ■ If alarms cannot be retrieved, phase displays **Connection error (SR0500.3.6.1)** error message (page 80). ■ If no error has occurred, phase displays the retrieved alarms at the top of the list of alarms.
Operator selects alarms for conversion	30	See **Convert alarms into exceptions (SR0500.2.3)** function (page 72).
Operator confirms phase	40	See **Confirm phase (SR0500.2.4)** function (page 73).

Retrieve alarms with automatic update (SR0500.2.2)

- Function: Automatically retrieve alarms from the DCS
- Trigger: Phase becomes active
- Postcondition: Phase is completed

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0500.1.2)** layout (page 68) with the list of retrieved alarms.
Phase retrieves alarms	20	Within the defined interval (**Automatic update (SR0500.8.4)** process parameter (page 75)), phase retrieves alarms and updates phase representation. ■ If alarms cannot be retrieved, phase displays the **Connection error (SR0500.3.6.1)** error message (page 80) (only if the phase's Active mode is the current view). The error message is only displayed once and does not stack for further occurrences.

Step	#	Description
		■ If there are new alarms, phase displays them at the top of the list of alarms and displays the **New alarms retrieved (SR0500.3.4.1)** information message (page 80) (only if the phase's Active mode is the current view). The information message is only displayed once and does not stack for further occurrences. ■ Retrieval is terminated when the operator confirms the phase (see step 50).
Operator taps **Get** button	30	Phase retrieves alarms from the DCS. ■ If alarms cannot be retrieved, phase displays **Connection error (SR0500.3.6.1)** error message (page 80). ■ If no error has occurred, phase displays the retrieved alarms at the top of the list of alarms.
Operator selects alarms for conversion	40	See **Convert alarms into exceptions (SR0500.2.3)** function (page 72).
Operator confirms phase	50	See **Confirm phase (SR0500.2.4)** function (page 73).

Convert alarms into exceptions (SR0500.2.3)

- Function: Retrieved alarms are converted into exceptions
- Trigger: Operator taps **Exception** button in **Active mode (SR0500.1.2)** layout (page 68) to convert selected alarms into exceptions
- Postcondition: Selected alarms are converted into exceptions and are no longer displayed in the list of alarms

Step	#	Description
Operator selects alarms for conversion	10	**Exception** button in **Active mode (SR0500.1.2)** layout (page 68) becomes active.
Operator taps **Exception** button in **Active mode (SR0500.1.2)** layout	20	Phase creates **Alarm exception (SR0500.3.1.1)** user-triggered exception (page 78). If selected alarms have already been converted by an operator within another Production Execution Client in the context of an event-triggered operation, phase displays **Alarms already converted (SR0500.3.6.2)** error message (page 81).

Confirm phase (SR0500.2.4)

- Function: Completion of phase
- Trigger: Operator confirms phase
- Postcondition: Phase is completed

Step	#	Description
Phase performs completion checks	10	- Phase polls DCS for new alarms. If there are new alarms, phase displays them at the top of the list of alarms and displays **New alarms retrieved (SR0500.3.4.1)** information message. - If alarms cannot be retrieved, phase creates **Retrieval exception (SR05003.2.1)** system-triggered exception (page 77). This exception has to be recorded only once. - Phase checks for not yet converted alarms only if the check has been enabled with the **Unconverted alarms (SR0500.8.6)** process parameter (page 76). If there are unconverted alarms, phase creates **Unconverted alarms (SR0500.3.2.2)** system-triggered exception (page 78). This exception has to be recorded each time unconverted alarms are detected.
Phase completion	20	Phase is completed.

Run several phases within an order (SR0500.2.6)

- Function: Several **Get DCS alarms** phases within the same order for the same target DCS
- Trigger: A subsequent phase instance becomes active
- Postcondition: N/A

Step	#	Description
Phase activation	10	Alarms that have already been converted by previous instances of the **Get DCS alarms** phase are not displayed again. Phase displays its user interface according to the **Active mode (SR0500.1.2)** layout (page 68) with the list of still unconverted alarms according to the shared filter criteria defined with the **Filter criteria (SR0500.8.3)** process parameter (page 74).

Process Parameters (SR0500.8+)

The following process parameters define the behavior of the phase.

BASIC PARAMETERS

Instruction (SR0500.8.1)

➢ For recent changes, see revision history (page 97).

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

DCS (SR0500.8.2)

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.

Filter criteria (SR0500.8.3)

Attribute	Type	Comment
Batch ID	String	Optional parameter to define the identifier of the batch running on the DCS. If not defined, phase displays "N/A".
Unit ID	String	Optional parameter to define the identifier of a unit of the DCS. 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".

Attribute	Type	Comment
Module IDs	Text (structured)	Optional parameter to define the list of equipment module IDs and control module IDs. If not defined, phase displays "N/A".

List editor (Framework capability)

The system provides a List editor for entering list items.

Automatic update (SR0500.8.4)

Attribute	Type	Comment
Enabled	Boolean	Controls if the query is repeated automatically. If so, ensure that the **Update interval** attribute is defined.
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.

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Retrieval exception (SR0500.8.5)

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.

See also **Retrieval exception (SR0500.3.2.1)** system-triggered exception (page 77).

Unconverted alarms (SR0500.8.6)

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

See also **Unconverted alarms (SR0500.3.2.2)** system-triggered exception (page 78).

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Record alarm exception (SR0500.8.7)

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.

See also **Alarm exception (SR0500.3.1.1)** user-triggered exception (page 78).

Exceptions (SR0500.3+)

The phase supports user-defined, user-triggered (page 78), system-triggered (page 77), and post-completion exceptions (page 80) and their configuration by means of process parameters (page 74).

User-defined exceptions cannot be configured by process parameters since they are provided by the framework and independent of phases.

System-triggered Exceptions (SR0500.3.2+)

➢ For recent changes, see revision history (page 97).

A system-triggered exception is represented in a message dialog along with an **Exception** button, in the Exception Window as the read-only description of the exception, and in the batch report.

The following system-triggered exceptions are available.

Retrieval exception (SR0500.3.2.1)

Representation of the exception:

- <Exception text>
  (taken from **Retrieval exception (SR0500.8.5)** process parameter (page 75))
  No alarms could be retrieved from the DCS.
  - Example:
    Issues with alarm retrieval
    No alarms could be retrieved from the DCS.

Retrieval exception - Logic (SR0500.3.2.1.1)

- Trigger: Alarms could not be retrieved from the DCS
- Postcondition: Exception is recorded

Step	#	Description
Operator accepts exceptional situation	10	Phase shows exception description to be signed.
Operator signs exception	20	Phase records exception.

Unconverted alarms (SR0500.3.2.2)

Representation of the exception:

- <Exception text>
  (taken from **Unconverted alarms (SR0500.8.6)** process parameter (page 76))
  Number of unconverted alarms: <value>
    - Example:
      Unconverted alarms
      Number of unconverted alarms: 3

Unconverted alarms - Logic (SR0500.3.2.2.1)

- Trigger: At least one alarm has not been converted to an exception at phase completion
- Postcondition: Exception is recorded

Step	#	Description
Operator accepts exceptional situation	10	Phase shows exception description to be signed.
Operator signs exception	20	Phase records exception. The unconverted alarms are no longer displayed in the list of alarms of the **Active mode (SR0500.1.2)** layout (page 68).

Multiple system-triggered exceptions (SR0500.3.2.3)

In case multiple system-triggered exceptions occur, only one combined exception (system-triggered exception) is recorded including information about all exceptions. The highest risk assessment of all related exceptions and its related signature privilege apply.

When the exception has been recorded, the phase is completed automatically.

User-triggered Exceptions (SR0500.3.1+)

A user-triggered exception is represented in the list of available user-triggered exceptions in the Exception Window, as the description of the exception, and in the batch report. The **Get DCS alarms** phase provides an **Exception** button in the Execution Window to record user-triggered exceptions.

The following user-triggered exceptions are available.

Alarm exception (SR0500.3.1.1)

The **Alarm exception** exception allows an operator to convert selected alarms from the DCS into phase-specific exceptions.

Representation of the exception:

- Exception text:
  <Exception text>
  (taken from **Record alarm exception (SR0500.8.7)** process parameter (page 76))
  <value> alarms have been converted.
  1st comment of exception:
  Timestamp: <timestamp of 1st alarm> / Source: <source of 1st alarm>
  Alarm: <1st alarm>
  Comment: <comment of 1st alarm, if available>
  :
  Last comment of exception:
  Timestamp: <timestamp of last alarm> / Source: <source of last alarm>
  Alarm: <last alarm>
  Comment: <comment of last alarm, if available>

  - Example:
    List of alarms converted into exceptions
    2 alarms have been converted.
    [Comment]Timestamp: 02/21/2016 14:35:29 PM CEST / Source: PV51525/ST-PV51525-001
    Alarm: Attr: LO_ALM Alarm Work: LOW State: ACT/UNACK Level: 15-CRITICAL Descript: Low Alarm Value 105 338 Limit 110
    Comment: Adjusted pressure by 4 psi oraita, 12:40:12 13-Nov-2015
    [Comment]Timestamp: 02/21/2016 14:36:12 PM CEST / Source: PV51525/ST-PV51525-001
    Alarm: Attr: LO_ALM Alarm Work: LOW State: INACT/UNACK Level: 15-CRITICAL Descript: Low Alarm Value 119 234 Limit 110

Alarm exception - Logic (SR0500.3.1.1.1)

- Trigger: Operator taps **Exception** button in **Active mode (SR0500.1.2)** layout (page 68) to convert selected alarms into exceptions
- Postcondition: Selected alarms are converted into exceptions and are no longer displayed in the list of alarms

Step	#	Description
Operator taps **Exception** button	10	Phase shows exception description to be signed.
Operator signs exception	20	Phase records exception and creates a comment per converted alarm. The converted alarms are no longer displayed in the list of alarms of the **Active mode (SR0500.1.2)** layout (page 68).

Post-completion Exceptions

There are no post-completion exceptions available.

Information Messages (SR0500.3.4+)

Information messages are represented in an information dialog containing a message type-specific icon, the information message, and an **OK** button.

The following information messages are available to inform the operator about how to proceed.

New alarms retrieved (SR0500.3.4.1)

UI text	Comment
There are <value> new alarms.	Message pack: PhaseDCSGetDCSAlarms<version> Message ID: NewAlarms_Msg
There is 1 new alarm.	Message pack: PhaseDCSGetDCSAlarms<version> Message ID: OneNewAlarm_Msg

Questions

There are no questions available.

Decisions

There are no decisions available.

Error Messages (SR0500.3.6+)

Error messages are represented in an error message dialog containing a message type-specific icon, the error message, and an **OK** button.

The following error messages are available to inform the operator about error conditions.

Connection error (SR0500.3.6.1)

UI text	Comment
Cannot retrieve alarms from the DCS.	Message pack: PhaseDCSGetDCSAlarms<version> Message ID: GetAlarms_Error

The **Details** button provides access to more specific technical information.

Alarms already converted (SR0500.3.6.2)

UI text	Comment
One or more of the selected alarms were already converted in another run of the phase.	Message pack: PhaseDCSGetDCSAlarms<version> Message ID: AlarmsAlreadyConverted_Error

Output Variables (SR0500.9+)

The following output variables are available to reference the phase's output.

Instance count (Framework capability)

- Data type: Long
- 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 (Framework capability)

- Data type: Timestamp
- Usage: The output variable provides the start time of the phase.

Completion time (Framework capability)

- Data type: Timestamp
- Usage: The output variable provides the completion time of the phase.

Identifier (Framework capability)

- Data type: String
- Usage: The output variable provides the identifier of the phase.

Number of retrieved alarms (SR0500.9.1)

- Data type: Long
- Usage: The output variable provides the number of alarms that were retrieved by the phase.

Number of converted alarms (SR0500.9.2)

- Data type: Long
- Usage: The output variable provides the number of alarms that were converted into exceptions.

Configuration Keys (SR0500.11+)

The following configuration keys are available to configure the phase's behavior.

Message broker URL (SR0500.11.1)

- **Phase/DCS/DCSMessageBrokerURL**
- **Type**: String
- **Value**: tcp://<hostname>:<port>
- **Description**: Specifies the URL of the messaging server used for DCS communication. Replace <hostname> by the name of the host to be used and <port> by the port number of the host to be used.
  If no value is set, the URL defined in the **MessageBrokerURL** configuration key is used.
- **Evaluated**: When a DCS phase communicates with a DCS.
- **Range**: N/A

Messaging timeout (SR0500.11.2)

- **Phase/DCS/DCSMessagingTimeout**
- **Type**: Long
- **Value**: 2
- **Description**: The messaging timeout for DCS communication in seconds.
  If no value is set, the timeout defined in the **MessagingTimeout** configuration key is used (converted into seconds).
- **Evaluated**: When a DCS phase communicates with a DCS.
- **Range**: N/A

Chapter 6

DCS Alarm-based Trigger Phase (SR0510+)

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.

Example use cases are:

- 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.

The configuration of the alarm query, number of retrieved alarms, fired triggers and their time information is stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 83).

Anomalies that occur during processing are covered by the phase exception handling (page 90) (e.g. timeout of the phase).

Layout

The phase provides a layout for its representation in the sub-report (page 83).

Representation during Execution

As a server-run phase, the phase has no graphical representation (UI).

Representation in Navigator

As a server-run phase, the phase is not visible in the Navigator.

Representation in Sub-report (SR0510.5+)

The sub-report contains the following information:

Common sub-report elements (Framework capability)

- <Start time>
- <Completion time>
- <Unit procedure> / <operation> / <phase>
- <Work center> / <station> / <device> - <phase completion user>

For phases running on a server, the phase completion-user corresponds to the system.

Sub-report elements (SR0510.5.1)

- DCS name: <logical name>
  Batch ID: <identifier>
  Unit ID: <identifier>
  Start: <timestamp>
  End: <timestamp>
  Modules: <list of identifiers>
- Number of retrieved alarms: <value>
- Number of executed triggers: <value>
- List of pause events: Paused from <timestamp> until <timestamp>

Business Logic (SR0510.2+)

The phase implements the following business logic.

Phase activation (SR0510.2.1)

- Function: Start the trigger processing
- Trigger: Phase becomes active
- Postcondition: Trigger processing is started

Step	#	Description
Phase activation	10	Phase checks ■ if the unit procedure is currently paused (check passes if the unit procedure is not paused), ■ if at least one of the relevant ETO templates is already active (check passes if at least one ETO template is active), and ■ if the timeout period has not elapsed yet (defined with the **Timeout period (SR0510.8.4)** process parameter (page 88)) (check passes if an ETO template has become active before the timeout period has elapsed). If all checks have passed, continue with step 10.4.
	10.1	If the unit procedure is paused, the phase waits without any action (no trigger processing, no timeout clock is running) until the unit procedure is continued.

Step	#	Description
	10.2	If the unit procedure is running, no relevant ETO template is active, and the timeout period has not elapsed, trigger processing is still waiting and timeout clock is running.
	10.3	If the timeout period has elapsed without any ETO becoming active, the phase is completed automatically according to the **Phase completion (SR0510.2.5)** function (page 86).
	10.4	If a relevant ETO template becomes active, trigger processing starts according to the **Fire triggers (SR0510.2.2)** function (page 85). At this point, the timeout period no longer applies.

Fire triggers (SR0510.2.2)

➢ For recent changes, see revision history (page 97).

- Function: Fire the triggers
- Trigger: Trigger processing has started
- Postcondition: Triggers are fired

Step	#	Description
Start trigger processing	10	Phase retrieves alarms from the DCS and starts to repeatedly retrieve alarms from the DCS according to the interval defined with the **Duration** attribute of the **Retrieving cycle (SR0510.8.3)** process parameter (page 88).
	20	Phase repeatedly fires a trigger, each time new alarms have been retrieved from the DCS.
DCS Adapter is not available	20.1	With respect to the alarms that could not be retrieved, phase creates a **Retrieval exception (SR0510.3.2.2)** system-triggered exception (page 91). As soon as the DCS Adapter becomes available again, phase retrieves the current alarms and continues to fire triggers according to the **Fire trigger (SR0510.2.2)** function. The system adds a comment to the already recorded exception according to the **Retrieval exception - Resume (SR0510.3.2.2.2)** function (page 91) of the **Retrieval exception (SR0510.3.2.2)** system-triggered exception (page 91).

Pause trigger processing (SR0510.2.3)

➢ For recent changes, see revision history (page 97).

- Function: Pause/continue the trigger processing
- Precondition: Trigger processing is active

- Trigger: Unit procedure is paused/continued by the operator (for details, see **Pausing a Unit Procedure (SR1089.8.3)** in "Functional Requirement Specification Execution Framework" [A1])
- Postcondition: Trigger processing is paused/continued

Step	#	Description
Operator pauses unit procedure	10	When the unit procedure of the phase is paused, trigger processing stops. No further triggers are fired automatically.
Operator continues paused unit procedure	20.1	When the paused unit procedure of the phase is continued, trigger processing continues according to the **Fire triggers (SR0510.2.2)** function (page 85).
	20.2	In case no relevant ETO template was active during the pause period and the timeout period has not elapsed yet: trigger processing is still waiting and the timeout clock is running, however, the timeout clock is reset upon resume of the pause.

Resume trigger processing (SR0510.2.4)

- Function: Resume the trigger processing
  (for the continuation of a paused unit procedure, see **Pause trigger processing (SR0510.2.3)** function (page 85))
- Precondition: Trigger processing is active
- Trigger: Phase is restarted, e.g. along with the restart of the operation that runs on the OES (for details, see **Resuming Server-run Operations (SR1200.1.3)** in "Functional Requirement Specification Execution Framework" [A1])
- Postcondition: Triggers are fired

Step	#	Description
Phase is restarted or DCS Adapter becomes available again	10	As soon as the phase is restarted or the DCS Adapter becomes available again, it retrieves the current alarms and continues to fire triggers according to the **Fire triggers (SR0510.2.2)** function (page 85).

Phase completion (SR0510.2.5)

- Function: Completion of phase
- Trigger: Timeout period has elapsed or no ETO template is active any more
- Postcondition: Phase is completed

Step	#	Description
Timeout period has elapsed	10	If the timeout period has elapsed, the phase is completed automatically without having fired any triggers and creates a **Timeout (SR0510.3.2.1)** system-triggered exception (page 90).
None of the previously running related ETO templates is active any more	20	Phase stops trigger processing and is completed automatically. If there are alarms for which no triggers have been fired, phase creates a **Post-ETO alarms (SR0510.3.2.3)** system-triggered exception (page 92).

Process Parameters (SR0510.8+)

The following process parameters define the behavior of the phase.

BASIC PARAMETERS

DCS (SR0510.8.1)

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.

Filter criteria (SR0510.8.2)

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

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" in the sub-report.
Module IDs	Text (structured)	Optional parameter to define the list of equipment module IDs and control module IDs. If not defined, phase displays "N/A" in the sub-report.

List editor (Framework capability)

The system provides a List editor for entering list items.

Retrieving cycle (SR0510.8.3)

Attribute	Type	Comment
Duration	Duration	Defines the interval between two consecutive reading actions. The minimum interval is one minute. If a value of less than one minute is set or the interval is not defined at all, the system interprets this as one minute.

Timeout period (SR0510.8.4)

Attribute	Type	Comment
Duration	Duration	Defines the duration that the phase waits for its event-triggered operation to become active, before the phase is automatically completed. Null is interpreted as 30 minutes (default timeout) during execution.

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Timeout exception (SR0510.8.5)

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.

See also **Timeout (SR0510.3.2.4)** system-triggered exception (page 90).

Retrieval exception (SR0510.8.6)

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.

See also **Retrieval exception (SR0510.3.2.2)** system-triggered exception (page 91).

Post-ETO alarms exception (SR0510.8.7)

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.

See also **Post-ETO alarms exception (SR0510.3.2.3)** system-triggered exception (page 92).

Exceptions (SR0510.3+)

The phase supports system-triggered exceptions (page 90) and their configuration by means of process parameters (page 87).

System-triggered Exceptions (SR0510.3.2+)

A system-triggered exception of a server-run phase is automatically recorded in the batch report without any user interaction.

The following system-triggered exceptions are available.

Timeout (SR0510.3.2.1)

In case the timeout period has elapsed, the system automatically records a system-triggered exception:

Representation of the exception:

- <Exception text>
(taken from **Timeout exception (SR0510.8.5)** process parameter (page 89))
Phase finished automatically due to timeout after <timeout period>.
- Example:
Timeout occurred.
Phase finished automatically due to timeout after 30 minutes.

Timeout - Logic (SR0510.3.2.1.1)

- Trigger: Phase is completed automatically due to timeout
- Postcondition: Exception is recorded

Step	#	Description
Timeout occurs	10	Phase automatically records exception without user interaction.

Retrieval exception (SR0510.3.2.2)

In case alarms from the DCS cannot be retrieved, the system automatically records a system-triggered exception:

Representation of the exception:

- <Exception text>
(taken from **Retrieval exception (SR0510.8.6)** process parameter (page 89))
Cannot connect to the DCS.
- Example:
Issues was alarm retrieval.
Cannot connect to the DCS.

Retrieval exception - Logic (SR0510.3.2.2.1)

- Trigger: Alarms could not be retrieved from the DCS
- Postcondition: Exception is recorded

Step	#	Description
Retrieval error occurs	10	Phase automatically records exception without user interaction.

Retrieval exception - Resume (SR0510.3.2.2.2)

- Trigger: Alarms can be retrieved from the DCS
- Postcondition: Comment is added to the corresponding exception

Step	#	Description
DCS Adapter is available again	10	As soon as the phase is able to retrieve alarms again, the system adds a comment to the exception (The connection to the DCS has been re-established.).

Post-ETO alarms exception (SR0510.3.2.3)

In case new alarms were retrieved from the DCS after the ETO template has been removed, the system cannot fire triggers and automatically records a system-triggered exception:

Representation of the exception:

- <Exception text>
  (taken from **Post-ETO alarms exception (SR0510.8.7)** process parameter (page 90))
  One or more DCS alarms have occurred after the last retrieval.
- Example:
  Alarms without trigger processing.
  One or more DCS alarms have occurred after the last retrieval.

Post-ETO alarms exception - Logic (SR0510.3.2.3.1)

- Trigger: Alarms were retrieved after the removal of the ETO template
- Postcondition: Exception is recorded

Step	#	Description
Alarms retrieved after removal of ETO template	10	Phase automatically records exception without user interaction and adds the affected alarms as comments to the exception.

Output Variables

The following output variables are available to reference the phase's output.

Instance count (Framework capability)

- Data type: Long
- 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 (Framework capability)

- Data type: Timestamp
- Usage: The output variable provides the start time of the phase.

Completion time (Framework capability)

- Data type: Timestamp
- Usage: The output variable provides the completion time of the phase.

Identifier (Framework capability)

- Data type: String
- Usage: The output variable provides the identifier of the phase.

Configuration Keys (SR0510.11+)

The following configuration keys are available to configure the phase's behavior.

Message broker URL (SR0510.11.1)

- **Phase/DCS/DCSMessageBrokerURL**
- **Type**: String
- **Value**: tcp://<hostname>:<port>
- **Description**: Specifies the URL of the messaging server used for DCS communication. Replace <hostname> by the name of the host to be used and <port> by the port number of the host to be used.
  If no value is set, the URL defined in the **MessageBrokerURL** configuration key is used.
- **Evaluated**: When a DCS phase communicates with a DCS.
- **Range**: N/A

Messaging timeout (SR0510.11.2)

- **Phase/DCS/DCSMessagingTimeout**
- **Type**: Long
- **Value**: 2
- **Description**: The messaging timeout for DCS communication in seconds.
  If no value is set, the timeout defined in the **MessagingTimeout** configuration key is used (converted into seconds).
- **Evaluated**: When a DCS phase communicates with a DCS.
- **Range**: N/A

Reference Documents

The following documents are available from the Rockwell Automation Download Site.

No.	Document Title	Part Number
A1	PharmaSuite Functional Requirement Specification Execution Framework	PSFRSEF-RM004E-EN-E
A2	Functional Requirement Specification DCS Adapter	DCFRSAD-RM001D-EN-E
A3	Technical Manual DCS Adapter	DCTMAD-GR001C-EN-E
A4	PharmaSuite Functional Requirement Specification Recipe and Workflow Management	PSFRSRD-RM008E-EN-E

TIP

To access the Rockwell Automation Download Site, you need to acquire a user account from Rockwell Automation Sales or Support.

Appendix A

Document Information

The document information covers various data related to the document.

Approval

This document has been approved electronically via the Rockwell Automation Document Management System (DMS). The required approvers of this document include the following:

Name	Role
Martin Dittmer	Product Manager
Steffen Landes	Development Manager
Martin Irmisch	Test Manager

In addition, the electronic document approval via DMS is confirmed by a handwritten signature of all approvers in the Quality Document when the release is completed. The Quality Document summarizes the quality-related planning activities and results of a PharmaSuite release.

Version Information

Object	Version
PharmaSuite	8.4
Create DCS batch	1.0 MR1
Get DCS batch values	1.0 MR1
Get DCS alarms	1.0 MR3
DCS alarm-based trigger	1.0 MR3
Functional Requirement Specification	1.1

Revision History

The following table describes the history of this document.

Changes related to the document:

Object	Description	Document
Document	The document applies to the **Create DCS batch** phase 1.0 MR1.	1.1

Changes related to "Automation Integration with DCS Phases" (page 3):

Object	Description	Document
---	---	---

Changes related to "Create DCS Batch" (page 7):

Object	Description	Document
Instruction (SR0520.8.1) (page 15)	Update The maximum length of the **Instruction** process parameter is 2000 characters (including HTML tags). No change of code.	1.0

Changes related to "Get DCS Batch Values" (page 35):

Object	Description	Document
System-triggered Exceptions (SR0525.3.2+) (page 52)	Update The message dialog of a system-triggered exception no longer provides a **Cancel** button.	1.0
Override DCS Batch Value - Logic (SR0525.3.1.1.1) (page 54)	Update Step 30.1: The message dialog no longer provides a **Cancel** button. Step 30.1.2: Phase requires to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).	1.0
Override DCS Batch Value - Logic (SR0525.3.1.2.1) (page 56)	Update Step 30.1: The message dialog no longer provides a **Cancel** button. Step 30.1.2: Phase requires to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).	1.0
Override DCS Batch Value - Logic (SR0525.3.1.3.1) (page 57)	Update Step 30.1: The message dialog no longer provides a **Cancel** button. Step 30.1.2: Phase requires to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).	1.0

Object	Description	Document
Instruction (SR0525.8.1) (page 45)	Update The maximum length of the **Instruction** process parameter is 2000 characters (including HTML tags). No change of code.	1.0

Changes related to "Get DCS Alarms" (page 67):

Object	Description	Document
Preview Mode (SR0500.1.1) (page 68)	Update The **Preview mode** layout displays update-related information.	1.0
Active Mode (SR0500.1.2) (page 68)	Update The **Active** mode layout displays update-related information.	1.0
System-triggered Exceptions (SR0500.3.2+) (page 77)	Update The message dialog of a system-triggered exception no longer provides a **Cancel** button.	1.0
Instruction (SR0500.8.1) (page 74)	Update The maximum length of the **Instruction** process parameter is 2000 characters (including HTML tags). No change of code.	1.0

Changes related to "DCS Alarm-based Trigger" (page 83):

Object	Description	Document
Fire Triggers (SR0510.2.2) (page 85)	Update Phase behavior when the DCS Adapter becomes available again: phase retrieves the current alarms and continues to fire triggers according to the **Fire triggers (SR0510.2.2)** function. No change of code.	1.0
Pause Trigger Processing (SR0510.2.3) (page 85)	Update Step 20.2 added: In case no relevant ETO template was active during the pause period and the timeout period has not elapsed yet: trigger processing is still waiting and the timeout clock is running, however, the timeout clock is reset upon resume of the pause. No change of code.	1.0

Index