FT PharmaSuite®

EBR PHASES
RELEASE 10.02.00
FUNCTIONAL REQUIREMENT SPECIFICATION

Rockwell Automation

Contact Rockwell See contact information provided in your maintenance contract.

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

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

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

Other Trademarks ActiveX, Microsoft, Microsoft Access, SQL Server, Visual Basic, Visual C++, Visual SourceSafe, Windows, Windows 7 Professional, Windows 10, Windows Server 2008, Windows Server 2012, and Windows Server 2016 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

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

ControlNet is a registered trademark of ControlNet International.

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

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

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

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

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

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

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

Contents

Figures

Introduction

This document details the requirements of the functions implemented by the phases specific to EBR. The phases are executed in the Production Execution Client of PharmaSuite.

Each requirement is composed of a name and a unique identifier (e.g., Instruction (SR0010.8.1)). If a requirement's meaning is for requirement grouping only, the identifier is appended by a plus sign (e.g., Process parameters (SR0010.8+)).
For requirements with **Framework capability** as identifier, see "Functional Requirement Specification Execution Framework" for their unique identifier, [A1] (page 179).

The revision history (page 182) list the changes made to the document with PharmaSuite 10.01.00 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.

Get Process Value Phase (SR0010+)

The **Get process value** phase allows an operator to capture process-related parameters.

Example use cases are:

- Manual entry of room temperature
  The room temperature must range between 20°C and 22°C. These boundary values can be defined as limits and corresponding limit violations can be tracked as exceptions.

- Recording of pH values
  The pH value of a material needs to be adjusted to a specified range. When capturing the pH value, the value can be checked against a specified range. Depending on the result, the building block can trigger loops within the recipe to further adjust the pH value.

- Recording of manually entered weighing values
  The relation between actual quantities and planned quantities is essential for the final product quality. Manually entered weighing values can be checked against three limit ranges (e.g. Warning limit, Control limit, Out of specification limit).

The process value can be entered manually during execution or can be populated as a default value from a previous phase.
The value is checked against configurable limits. The phase supports up to three limit ranges.

The recorded value is stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 5).
Anomalies that occur during processing are covered by the phase exception handling (page 15) (e.g. limit violation).

After completion the phase displays the recorded value, both in the Execution Window and the Navigator. Additionally, the Navigator provides access to the post-completion exceptions.

Fill and run the tablet press. Monitor the pressure.	Displayed pressure	6.5 t
	Upper destruction limit	13.0 t
	Upper warning limit	10.0 t
	Upper attention limit	8.0 t
	Lower attention limit	5.5 t
	Lower warning limit	5.0 t Confirm
	Lower destruction limit	4.5 t

Figure 1: Get process value during execution

Layout

The phase provides individual layouts for its representation during execution (page 4), in the Navigator (page 5), and in the sub-report (page 5).

Representation during Execution (SR0010.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0010.1.2)

1. <Instruction text> (column 1 and column 2)
   (taken from **Instruction (SR0010.8.1)** process parameter (page 9))

2. Box for <actual process value, default value>
   (UoM taken from **Value configuration (SR0010.8.2)** process parameter (page 9) and default taken from **Limit definition (SR0010.8.5)** process parameter (page 12))

3. List of configured limits
   (taken from **Limit configuration (SR0010.8.4)** process parameter (page 10) and **Limit definition (SR0010.8.5)** process parameter (page 12))

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

Active mode (SR0010.1.1)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text> (column 1 and column 2)
   (taken from **Instruction (SR0010.8.1)** process parameter (page 9))

3. Box for <actual process value, default value><UoM>
   (UoM and editable status taken from **Value configuration (SR0010.8.2)** process parameter (page 9) and default taken from **Limit definition (SR0010.8.5)** process parameter (page 12))

4. List of configured limits
   (taken from **Limit configuration (SR0010.8.4)** process parameter (page 10) and **Limit definition (SR0010.8.5)** process parameter (page 12))

5. **Confirm** button.

Completed mode (SR0010.1.3)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text> (column 1 and column 2)
   (taken from **Instruction (SR0010.8.1)** process parameter (page 9))

3. <Actual value>
   (UoM taken from **Value configuration (SR0010.8.2)** process parameter (page 9))

4. List of configured limits
   (taken from **Limit configuration (SR0010.8.4)** process parameter (page 10) and
   Limit definition (SR0010.8.5) process parameter (page 12))

5. **Confirm** button (completed).

Representation in Navigator (SR0010.4+)

The Navigator provides the following details:

Phase column (Framework capability)

■ <Phase name>

  ■ Example:
    Read processing value

Information column (SR0010.4.1)

■ <Actual value> <UoM>

  ■ Example: 41%

Action column (SR0010.4.2)

■ Correct, provides exception to correct the recorded value.

Representation in Sub-report (SR0010.5+)

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 (SR0010.5.1)

- Instruction table panel and/or instruction link panel
(only if an instruction table and/or instruction link is defined for the phase)

- Two instruction texts

- Actual value with UoM

- Limit information

Business Logic (SR0010.2+)

The phase implements the following business logic.

Document process value (SR0010.2.1)

- Function: Document a process value

- Trigger: Phase becomes active

- Postcondition: Process value is documented

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0010.1.1)** layout (page 4).
	20	- If no default value is set, operator enters process value. - If a default value is set and the default value is editable, operator accepts default value or enters process value. - If a default value is set and the default value is not editable, operator accepts default value.
Cursor leaves box	30	Phase triggers **Validate process value (SR0010.2.2)** function (page 7).

Validate process value (SR0010.2.2)

- Function: Validate a process value

- Trigger: Cursor leaves the box that holds the actual process value

- Postcondition: Process value is validated

Step	#	Description
Validation	10	Phase checks the value against the settings of the **Limit definition (SR0010.8.5)** process parameter (page 12). Limits are checked in the following order: LLL/HHH » LL/HH » L/H.
	10.1	If the check is violated, phase creates the **Limit violation (SR0010.3.2.1)** system-triggered exception (page 15).
	10.2	If the check is not violated, phase is completed.

Process Parameters (SR0010.8+)

The following process parameters define the behavior of the phase.

INSTRUCTION TABLE-SPECIFIC PARAMETERS

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

INSTRUCTION LINK-SPECIFIC PARAMETERS

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 8). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.

Attribute	Type	Comment
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

BASIC PARAMETERS

Instruction (SR0010.8.1)

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

PROCESS VALUE PARAMETERS

Value configuration (SR0010.8.2)

Attribute	Type	Comment
UoM	Unit of measure	Must match a unit of measure available within PharmaSuite. See also attributes of the **Limit definition (SR0010.8.5)** process parameter (page 12).
Value editable	Flag	Controls if the displayed value is editable during execution. Default setting: **Yes**

> **TIP**
>
> 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.

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Limit configuration (SR0010.8.4)

During execution, the actual process value is checked against the configured limits when the cursor leaves the box that holds the actual process value. If the checks are activated for the available limit ranges, the checks are performed in the following order:

1. LLL-HHH
2. LL-HH
3. L-H.

L-H configuration

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **L limit** and **H limit** attributes of the **Limit definition** process parameter (page 12) are set.
Display	Flag	Controls if the limit range is displayed during execution.
Lower limit name	Text	Defines the name of the lower limit displayed during execution.
Upper limit name	Text	Defines the name of the upper limit displayed during execution.
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 2000 characters.

LL-HH configuration

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **LL limit** and **HH limit** attributes of the **Limit definition** process parameter (page 12) are set.
Display	Flag	Controls if the limit range is displayed during execution if the check is enabled.
Lower limit name	Text	Defines the name of the lower limit displayed during execution.
Upper limit name	Text	Defines the name of the upper limit displayed during execution.
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 2000 characters.

LLL-HHH configuration

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **LLL limit** and **HHH limit** attributes of the **Limit definition** process parameter (page 12) are set.
Display	Flag	Controls if the limit range is displayed during execution if the check is enabled.
Lower limit name	Text	Defines the name of the lower limit displayed during execution.
Upper limit name	Text	Defines the name of the upper limit displayed during execution.

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 2000 characters.

See also **Limit violation (SR0010.3.2.1)** system-triggered exception (page 15).

Limit definition (SR0010.8.5)

The following rules apply to the attributes:

- The unit of measure must be of the same system of measurement as the one used for the **Value configuration** process parameter (page 9) (e.g. weight: mg, kg, pound; length: mm, m, inch).

- LLL limit < LL limit < L limit < Reference value < H limit < HH limit < HHH limit

Attribute	Type	Comment
LLL limit	MeasuredValue	Define 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.
LL limit	MeasuredValue	
L limit	MeasuredValue	
Reference value	MeasuredValue	Defines the reference value in case of a limit range of the **Relative** limit type.
H limit	MeasuredValue	Define the values of the upper limits (including the values themselves).
HH limit	MeasuredValue	

Attribute	Type	Comment
HHH limit	MeasuredValue	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-H type	Choice list	Define the type of the limit range (**Absolute**, **Relative**).
LL-HH type	Choice list	During execution, the phase always calculates and displays absolute values.
LLL-HHH type	Choice list	Default setting: **Absolute**.
Default value	MeasuredValue	Defines the default value.

The following limit types are available: **Absolute** and **Relative**. The limits are calculated according to the following definitions.

Limit	Absolute value definition	Relative value definition
HHH limit	HHH	Reference value + HHH
HH limit	HH	Reference value + HH
H limit	H	Reference value + H
L limit	L	Reference value - L
LL limit	LL	Reference value - LL
LLL limit	LLL	Reference value - LLL

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Override value (SR0010.8.7)

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 2000 characters.

See also **Override value (SR0010.3.1.1)** user-triggered exception (page).

CONFIGURATION OF POST-COMPLETION EXCEPTIONS

Correct value (SR0010.8.6)

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.**
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

See also **Correct value (SR0010.3.3.1)** post-completion exception (page).

Exceptions (SR0010.3+)

The phase supports user-defined, user-triggered (page 16), system-triggered (page 15), and post-completion exceptions (page 17) and their configuration by means of process parameters (page 7).

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

System-triggered Exceptions (SR0010.3.2+)

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 (SR0010.3.2.1)

Representation of the exception:

■ <Exception text>
(taken from **Limit configuration (SR0010.8.4)** process parameter (page 10))
<Limit name>: <expected value>
Actual value: <process value>

■ Example:
Limit violation confirmed.
Lower warning: 300 rpm.
Actual value: 200 rpm

Limit violation - Completion (SR0010.3.2.2)

■ Trigger: Process value is not within the defined limits

■ Postcondition: N/A

Step	#	Description
Operator triggers exception	10	Phase records the exception.
	20	In case of a limit violation, the phase can be completed if all of the following applies: ■ The respective exception was recorded. ■ The value has not been changed again after the exception was recorded.

User-triggered Exceptions (SR0010.3.1+)

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.

Override value (SR0010.3.1.1)

The **Override value** exception allows an operator to override the value in case it is set to **read-only** (**Value editable** attribute of the **Value configuration (SR0010.8.2)** process parameter (page 9)).

Representation during exception handling:

- Instruction:
  Please enter new value.
  <Old value with unit of measure>
  Box for new value (with unit of measure)
  Confirm button.

- <Exception text>
  (taken from **Override value (SR0010.8.7)** process parameter (page 14))
  Old value: <old value> <UoM>
  New value: <new value> <UoM>

  - Example:
    Speed value corrected.
    Old value: 20 rpm
    New value: 25 rpm

Override value - Logic (SR0010.3.1.1.1)

- Trigger: Exception is selected

- Postcondition: Value is overridden

Step	#	Description
Operator triggers exception	10	Phase displays Exception Window.
	20	Operator enters new value.
Operator confirms exception	30	Phase shows exception description to be signed according to **Override value (SR0010.8.7)** process parameter (page 14).
Operator signs exception	40	Phase records the exception.

Post-completion Exceptions (SR0010.3.3+)

A post-completion exception is accessible via the Navigator and represented in the list of available post-completion exceptions in the Exception Window, as the description of the exception, and in the batch report.

The following post-completion exceptions are available.

Correct value (SR0010.3.3.1)

The **Correct value** exception allows an operator to correct the recorded value from the Navigator after the completion of the phase.

> **TIP**
>
> A recorded value could be used within branching. The correction of a value **does not influence** already processed branching decisions.

Representation of the exception:

- Instruction:
  Please enter new value.
  <Old value with unit of measure>
  Box for new value (with unit of measure)
  Confirm button.

- <Exception text>
  (taken from **Correct value (SR0010.8.6)** process parameter (page 14))

  - Example:
    Speed value corrected.

Correct value - Validation (SR0010.3.3.2)

- Trigger: Phase is completed

- Postcondition: Value is corrected

Step	#	Description
Operator triggers action	10	Phase displays Exception Window.
	20	Operator enters corrected value.
Cursor leaves box	30	Phase checks the value against the settings of the **Limit definition (SR0010.8.5)** process parameter (page 12).
	30.1	See **Correct value - Logic 2 (SR0010.3.3.3)**.
	30.2	If the limit is not violated, the corrected value is documented.

Correct value - Combined exception (SR0010.3.3.3)

- Trigger: Limit is violated

- Postcondition: Post-completion exception is recorded

Step	#	Description
Limit is violated	10	If the limit is violated, only one exception (post-completion exception) is recorded including information about all related exceptions. The highest risk assessment of all related exceptions and its related signature privilege apply.

Information Messages

There are no information messages available.

Questions

There are no questions available.

Decisions

There are no decisions available.

Error Messages

There are no error messages available.

Output Variables (SR0010.9+)

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.

Value (SR0010.9.4)

- Data type: MeasuredValue

- Usage: The output variable provides the complete process value as a **MeasuredValue** object.

Unit of measure (SR0010.9.3)

- Data type: String

- Usage: The output variable provides the unit of measure of the process value.

Get Text Value Phase (SR0020+)

The **Get text value** phase allows an operator to record text (a string) during execution.

Example use cases are:

■ Recording of visual appearance during product test
During the inspection of a product sample, the visual appearance of the sample can be documented (e.g. transparent, cloudy).

■ Checking the expected representation of a recipe at an equipment unit
On the display of a piece of equipment, the machine recipe is visualized. When setting up the equipment unit, the quality of the recipe's representation can be checked against an expected string (e.g. difficult to read).

■ Recording of production resources
Operator documents which tool was used when entering the property tag.

The text can be entered manually during execution or can be populated as a default text from a previous phase.
The text is checked against configurable text.

The recorded value is stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 23).
Anomalies that occur during processing are covered by the phase exception handling (page 30) (e.g. limit violation).

After completion the phase displays the recorded value, both in the Execution Window and the Navigator. Additionally, the Navigator provides access to the post-completion exception.
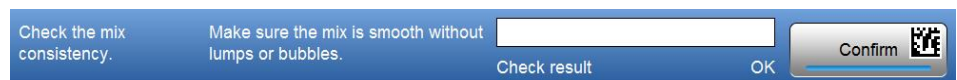

Figure 2: Get text value during execution

Layout

The phase provides individual layouts for its representation during execution (page 22), in the Navigator (page 23), and in the sub-report (page 23).

Representation during Execution (SR0020.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0020.1.2)

1. <Instruction text> (column 1 and column 2)
   (taken from **Instruction (SR0020.8.1)** process parameter (page 27))

2. Box for <actual text value, default value>
   (Default taken from **Expected value definition (SR0020.8.5)** process parameter (page 28))

3. Configured expected value
   (taken from **Expected value configuration (SR0020.8.4)** process parameter (page 28) and **Expected value definition (SR0020.8.5)** process parameter (page 28))

4. **Confirm** button (completed).

Active mode (SR0020.1.1)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text> (column 1 and column 2)
   (taken from **Instruction (SR0020.8.1)** process parameter (page 27))

3. Box for <actual text value, default value>
   (Default taken from **Expected value definition (SR0020.8.5)** process parameter (page 28))

4. Configured expected value
   (taken from **Expected value configuration (SR0020.8.4)** process parameter (page 28) and **Expected value definition (SR0020.8.5)** process parameter (page 28))

5. **Confirm** button.

Completed mode (SR0020.1.3)

1. Instruction table panel and/or instruction link panel
(only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text> (column 1 and column 2)
(taken from **Instruction (SR0020.8.1)** process parameter (page 27))

3. <Actual value>

4. Configured expected value
(taken from **Expected value configuration (SR0020.8.4)** process parameter
(page 28) and **Expected value definition (SR0020.8.5)** process parameter (page
28))

5. **Confirm** button (completed).

Representation in Navigator (SR0020.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>

  - Example:
    Read processing value

Information column (SR0020.4.1)

- <Actual value>

  - Example: Not OK

Action column (SR0020.4.2)

- Correct, provides exception to correct the recorded value.

Representation in Sub-report (SR0020.5+)

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 (SR0020.5.1)

- Instruction table panel and/or instruction link panel
  (only if an instruction table and/or instruction link is defined for the phase)

- Two instruction texts

- Actual value

- Expected value information

Business Logic (SR0020.2+)

The phase implements the following business logic.

Document text value (SR0020.2.1)

- Function: Document a text value

- Trigger: Phase becomes active

- Postcondition: Text value is documented

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0020.1.1)** layout (page 22).
	20	- If no default value is set, operator enters text value. - If a default value is set and the default value is editable, operator accepts default value or enters text value. - If a default value is set and the default value is not editable, operator accepts default value.
Cursor leaves box	30	Phase triggers **Validate text value (SR0020.2.2)** function (page 25).

Validate text value (SR0020.2.2)

- ■ Function: Validate a text value

- ■ Trigger: Cursor leaves the box that holds the actual text value

- ■ Postcondition: Text value is validated

Step	#	Description
Validation	10	Phase checks the value against the settings of the **Expected value definition (SR0020.8.5)** process parameter (page 28).
	10.1	If the check is violated, phase creates the **Violation of expected value (SR0020.3.2.1)** system-triggered exception (page 30).
	10.2	If the check is not violated, phase is completed.

Process Parameters (SR0020.8+)

The following process parameters define the behavior of the phase.

INSTRUCTION TABLE-SPECIFIC PARAMETERS

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

INSTRUCTION LINK-SPECIFIC PARAMETERS

Instruction text with links (Framework capability)

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

Instruction link definition (Framework capability)

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

BASIC PARAMETERS

Instruction (SR0020.8.1)

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

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Expected value configuration (SR0020.8.4)

During execution, the actual text value is checked against the configured limits when the cursor leaves the box that holds the actual process value.

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **Expected value** attribute of the **Expected value definition (SR0020.8.5)** process parameter (page 28) is set.
Display	Flag	Controls if an expected value is displayed during execution.
Expected value name	Text	Defines the name of the expected value.
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 2000 characters.

See also **Violation of expected value (SR0020.3.2.1)** system-triggered exception (page 30).

Expected value definition (SR0020.8.5)

Attribute	Type	Comment
Expected value	Text	Defines the expected value. Maximum length is 256 characters.
Default value	Text	Defines the default value. Maximum length is 256 characters.
Value editable	Flag	Controls if the displayed value is editable during execution. Default setting: **Yes**

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Override value (SR0020.8.7)

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 2000 characters.

See also **Override value (SR0020.3.1.1)** user-triggered exception (page 31).

CONFIGURATION OF POST-COMPLETION EXCEPTIONS

Correct value (SR0020.8.6)

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.**
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

See also **Correct value (SR0020.3.3.1)** post-completion exception (page 32).

Exceptions (SR0020.3+)

The phase supports user-defined, user-triggered (page 31), system-triggered (page 30), and post-completion exceptions (page 32) and their configuration by means of process parameters (page 25).

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

System-triggered Exceptions (SR0020.3.2+)

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.

Violation of expected value (SR0020.3.2.1)

Representation of the exception:

- \<Exception text\>
  (taken from **Expected value configuration (SR0020.8.4)** process parameter (page 28))
  Expected value: \<expected value\>
  Actual value: \<text value\>

- Example:
  Expected value violation confirmed.
  Expected value: OK
  Actual value: Not Ok

Violation of expected value - Completion (SR0020.3.2.2)

- Trigger: Text value deviates from the defined expected value

- Postcondition: N/A

Step	#	Description
Operator triggers exception	10	Phase records the exception.
	20	In case of a violation of an expected value, the phase can be completed if all of the following applies: ■ The respective exception was recorded. ■ The value has not been changed again after the exception was recorded.

User-triggered Exceptions (SR0020.3.1+)

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.

Override value (SR0020.3.1.1)

The **Override value** exception allows an operator to override the value in case it is set to **read-only** (**Value editable** attribute of the **Expected value definition (SR0020.8.5)** process parameter (page 28)).

Representation during exception handling:

- Instruction:
  Please enter new value.
  <Old value>
  Box for new value
  Confirm button.

- <Exception text>
  (taken from **Override value (SR0020.8.7)** process parameter (page 29))
  Old value: <old value>
  New value: <new value>

  - Example:
    Used recipe number corrected.
    Old value: R2011_v1
    New value: R2011_v2

Override value - Logic (SR0020.3.1.1.1)

- Trigger: Exception is selected

- Postcondition: Value is overridden

Step	#	Description
Operator triggers exception	10	Phase displays Exception Window.
	20	Operator enters new value.
Operator confirms exception	30	Phase shows exception description to be signed according to **Override value (SR0020.8.7)** process parameter (page 29).
Operator signs exception	40	Phase records the exception.

Post-completion Exceptions (SR0020.3.3+)

A post-completion exception is accessible via the Navigator and represented in the list of available post-completion exceptions in the Exception Window, as the description of the exception, and in the batch report.

The following post-completion exceptions are available.

Correct value (SR0020.3.3.1)

The **Correct value** exception allows an operator to correct the recorded value from the Navigator after the completion of the phase.

> **TIP**
>
> A recorded value could be used within branching. The correction of a value **does not influence** already processed branching decisions.

Representation of the exception:

- Instruction:
  Please enter new value.
  \<Old value\>
  Box for new value
  Confirm button.

- \<Exception text\>
  (taken from **Correct value (SR0020.8.6)** process parameter (page 29))

  - Example:
    Check result corrected.

Correct value - Validation (SR0020.3.3.2)

- Trigger: Phase is completed

- Postcondition: Value is corrected

Step	#	Description
Operator triggers action	10	Phase displays Exception Window.
	20	Operator enters corrected value.
Cursor leaves box	30	Phase checks the value against the settings of the **Expected value definition (SR0020.8.5)** process parameter (page 28).
	30.1	See **Correct value - Logic 2 (SR0020.3.3.3)**.
	30.2	If the limit is not violated, the corrected value is documented.

Correct value - Combined exception (SR0020.3.3.3)

- ■ Trigger: Limit is violated

- ■ Postcondition: Post-completion exception is recorded

Step	#	Description
Limit is violated	10	If the limit is violated, only one exception (post-completion exception) is recorded including information about all related exceptions. The highest risk assessment of all related exceptions and its related signature privilege apply.

Information Messages

There are no information messages available.

Questions

There are no questions available.

Decisions

There are no decisions available.

Error Messages

There are no error messages available.

Output Variables (SR0020.9+)

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.

Value (SR0020.9.2)

- Data type: String

- Usage: The output variable provides the text value entered during execution as string of characters.

Show Instruction Text (SR0030+)

The **Show instruction text** phase allows to display a specific instruction related to the process step the operator is executing.

Example use cases are:

- Description of how to assemble equipment.

- Description of how to sample the product.

- Information about specifics of a process activity.

The instruction text is stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 37).

Figure 3: Show instruction text during execution

Layout

The phase provides individual layouts for its representation during execution (page 35), in the Navigator (page 36), and in the sub-report (page 37).

Representation during Execution (SR0030.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0030.1.2)

1. <Instruction text> (column 1, column 2, and column 3)
(taken from **Instruction (SR0030.8.1)** process parameter (page 40) and the **Layout (SR0030.8.2)** process parameter (page 40))

2. **Confirm** button (disabled).

Active mode (SR0030.1.1)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text> (column 1, column 2, and column 3)
   (taken from **Instruction (SR0030.8.1)** process parameter (page 40) and the
   Layout (SR0030.8.2) process parameter (page 40))

3. **Confirm** button.

Completed mode (SR0030.1.3)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text> (column 1, column 2, and column 3)
   (taken from **Instruction (SR0030.8.1)** process parameter (page 40) and the
   Layout (SR0030.8.2) process parameter (page 40))

3. **Confirm** button (completed).

Representation in Navigator (SR0030.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>
  - Example:
    Mix inputs

Information column (SR0030.4.1)

- <Empty>

Action column

- There are no phase-specific actions available.

Representation in Sub-report (SR0030.5+)

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 (SR0030.5.1)

- ◼ Instruction table panel and/or instruction link panel
  (only if an instruction table and/or instruction link is defined for the phase)

- ◼ Up to three instruction texts

Business Logic (SR0030.2+)

The phase implements the following business logic.

Display instruction text (SR0030.2.1)

- ◼ Function: Display an instruction text

- ◼ Trigger: Phase becomes active

- ◼ Postcondition: Instruction text was displayed

Step	#	Description
Phase activation	10	Phase displays instruction text.

Process Parameters (SR0030.8+)

The following process parameters define the behavior of the phase.

INSTRUCTION TABLE-SPECIFIC PARAMETERS

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

INSTRUCTION LINK-SPECIFIC PARAMETERS

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 39). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

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

BASIC PARAMETERS

Instruction (SR0030.8.1)

Attribute	Type	Comment
Column 1	HTML text	Instruction text to be displayed. **Restriction**: Maximum length is 2000 characters (including HTML tags). The layout settings define which columns will be visible (see **Type** attribute of the **Layout (SR0030.8.2)** process parameter (page 40)).
Column 2	HTML text	
Column 3	HTML text	

Layout (SR0030.8.2)

Attribute	Type	Comment
Type	Choice list	Defines the layout of the column(s) holding the instruction texts: **1** column, **2** columns (with narrow first column and wide second column), **2** columns (with wide first column and narrow second column), or **3** columns). Default setting: **1 column**.

Exceptions

The phase supports user-defined, user-triggered (page 41), system-triggered (page 41), and post-completion exceptions (page 41) and their configuration by means of process parameters (page 38).

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

There are no user-triggered exceptions available.

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

There are no error messages available.

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.

Show Document (SR0040+)

The **Show document** phase allows to display a variety of documents.

Example use cases are:

■ Display an SOP
An SOP is stored within a central DMS. This very SOP can be displayed to the operator during execution.

■ Show an instruction video
The instruction of a GMP-critical process step is available as a video. The video can be shown during execution.

The name of the work instruction (document) and the instruction text are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 45).

The name of the displayed document is shown on the detail information button in the Navigator (e.g. SOP).
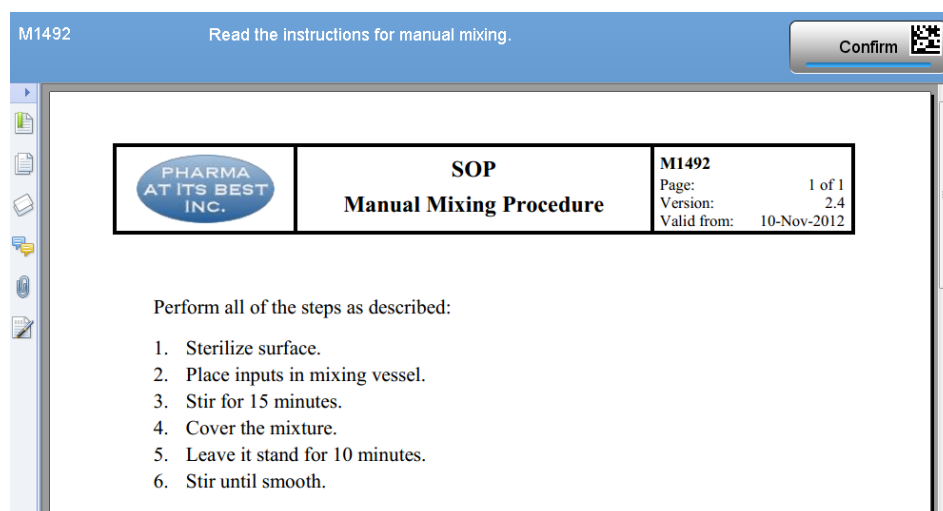
Figure 4: Show document during execution

Layout

The phase provides individual layouts for its representation during execution (page 44), in the Navigator (page 45), and in the sub-report (page 45).

Representation during Execution (SR0040.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0040.1.2)

1. \<Name of the work instruction\>
   (taken from **Document (SR0040.8.2)** process parameter (page 48))

2. \<Instruction text\>
   (taken from **Instruction (SR0040.8.1)** process parameter (page 48))

3. **Confirm** button (disabled).

Active mode (SR0040.1.1)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. \<Name of the work instruction\>
   (taken from **Document (SR0040.8.2)** process parameter (page 48))

3. \<Instruction text\>
   (taken from **Instruction (SR0040.8.1)** process parameter (page 48))

4. \<Work instruction\>, in an HTML container

5. **Confirm** button.

Completed mode (SR0040.1.3)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. \<Name of the work instruction\>
   (taken from **Document (SR0040.8.2)** process parameter (page 48))

3. \<Instruction text\>
   (taken from **Instruction (SR0040.8.1)** process parameter (page 48))

4. **Confirm** button (completed).

Representation in Navigator (SR0040.4+)

The Navigator provides the following details:

Phase column (Framework capability)

■ <Phase name>

    ■ Example:
       Mix inputs

Information column (SR0040.4.1)

■ <Name of the work instruction>

    ■ Example: SOP

Action column

■ There are no phase-specific actions available.

Representation in Sub-report (SR0040.5+)

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 (SR0040.5.1)

■ Instruction table panel and/or instruction link panel
(only if an instruction table and/or instruction link is defined for the phase)

■ Name of the work instruction

■ Instruction text

Business Logic (SR0040.2+)

The phase implements the following business logic.

Display document (SR0040.2.1)

- ■ Function: Display a document

- ■ Trigger: Phase becomes active

- ■ Postcondition: Document was displayed

Step	#	Description
Phase activation	10	Phase displays document.

Process Parameters (SR0040.8+)

The following process parameters define the behavior of the phase.

INSTRUCTION TABLE-SPECIFIC PARAMETERS

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

INSTRUCTION LINK-SPECIFIC PARAMETERS

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 47). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.

Attribute	Type	Comment
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

BASIC PARAMETERS

Instruction (SR0040.8.1)

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

Document (SR0040.8.2)

Attribute	Type	Comment
Work instruction	Text	Name of a FactoryTalk ProductionCentre work instruction object. The document will be shown within an HTML container.

Exceptions

The phase supports user-defined, user-triggered (page 48), system-triggered (page 48), and post-completion exceptions (page 49) and their configuration by means of process parameters (page 46).

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

There are no user-triggered exceptions available.

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

There are no error messages available.

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.

Get Choice Value Phase (SR0080+)

The **Get choice value** phase allows an operator to record a choice from a pre-defined list of options.

Example use cases are:

■ Recording of visual appearance during product test
During the inspection of a product sample, the visual appearance of the sample can be selected from a pre-defined list (e.g. Transparent, Cloudy, Dark).

■ Recording of production resources from a pre-defined list with a preset default option
Operator documents which tool was used when entering the property tag.

■ Recording of an operator decision
Operator documents with **Yes** or **No** whether a certain precondition applies. The operator decision determines which of the two alternative subsequent process steps of a selection branch will become active.

The selected option is checked against configurable options.

The recorded value is stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 53).
Anomalies that occur during processing are covered by the phase exception handling (page 59) (e.g. deviation).

After completion the phase displays the selected option in the Execution Window. The Navigator displays the selected option and provides access to the post-completion exception.
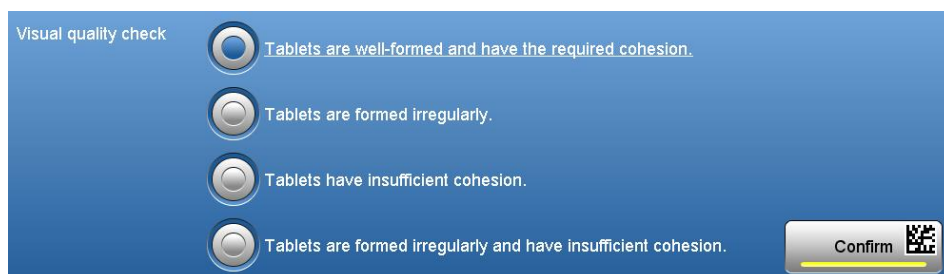
Figure 5: Get choice value during execution

Layout

The phase provides individual layouts for its representation during execution (page 52), in the Navigator (page 53), and in the sub-report (page 53).

Representation during Execution (SR0080.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0080.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0080.8.1)** process parameter (page 57))

2. List of options
   (taken from **List of options (SR0080.8.2)** process parameter (page 57))
   In case of configuration errors, phase displays **Invalid expected value configuration (SR0080.3.6.1)** error message (page 63), **Invalid default value configuration (SR0080.3.6.2)** error message (page 63), or **Invalid choice item configuration (SR0080.3.6.3)** error message (page 63).

3. **Confirm** button (disabled).

Active mode (SR0080.1.2)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
   (taken from **Instruction (SR0080.8.1)** process parameter (page 57))

3. List of options
   (taken from **List of options (SR0080.8.2)** process parameter (page 57))
   In case of configuration errors, phase displays **Invalid expected value configuration (SR0080.3.6.1)** error message (page 63), **Invalid default value configuration (SR0080.3.6.2)** error message (page 63), or **Invalid choice item configuration (SR0080.3.6.3)** error message (page 63).

4. **Confirm** button.

Completed mode (SR0080.1.3)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
   (taken from **Instruction (SR0080.8.1)** process parameter (page 57))

3. Selected option

4. **Confirm** button (completed).

Representation in Navigator (SR0080.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- ■ <Phase name>
  - ■ Example:
    Get visual appearance

Information column (SR0080.4.1)

- ■ <Selected option>
  - ■ Example: Initially cloudy

Action column (SR0080.4.2)

- ■ Correct, provides exception to correct the selected option.

Representation in Sub-report (SR0080.5+)

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 (SR0080.5.1)

- ■ Instruction table panel and/or instruction link panel
  (only if an instruction table and/or instruction link is defined for the phase)
- ■ Instruction text
- ■ List of options
  - ■ Selected
  - ■ Expected
  - ■ Key
  - ■ Text

Business Logic (SR0080.2+)

The phase implements the following business logic.

Display choice values (SR0080.2.1)

- Function: Display of choice values

- Trigger: Phase becomes active

- Postcondition: Phase is active

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0080.1.2)** layout (page 52).
	20	In case of configuration errors, phase displays **Invalid expected value configuration (SR0080.3.6.1)** error message (page 63), **Invalid default value configuration (SR0080.3.6.2)** error message (page 63), or **Invalid choice item configuration (SR0080.3.6.3)** error message (page 63).

Select choice value (SR0080.2.2)

- Function: Choice value selection

- Trigger: Operator selects a choice value

- Postcondition: Phase is active

Step	#	Description
Operator selects choice value	10	Phase marks selected choice value as selected.
Phase checks setting of **One-click completion (SR0080.8.4)** process parameter (page 58)	20	One-click completion is enabled: ■ Phase is completed automatically. ■ Phase checks expected value in corresponding step of **Confirm phase (SR0080.2.3)** function (page 55). If a **system-triggered (SR0080.3.2+)** exception (page 60) occurs during phase completion, phase is not completed after the exception has been signed. Phase returns to the **Active mode (SR0080.1.2)** layout (page 52) and the operator can complete the phase with the **Confirm** button.

Confirm phase (SR0080.2.3)

- ■ Function: Completion of phase

- ■ Trigger: Operator confirms phase

- ■ Postcondition: Phase is completed

Step	#	Description
Operator confirms phase	10	Operator confirms choice value.
Selection check	15	If no value has been selected, phase displays the **No choice item selected (SR0080.3.6.4)** error message (page 63). When the error message has been confirmed, phase returns to the **Active mode (SR0080.1.2)** layout (page 52). Otherwise continue with step 20.
Validation	20	Phase checks the value against the settings of the **Expected value definition (SR0080.8.3)** process parameter (page 59).
	20.1	If the check is violated, phase creates the **Expected value check (SR0080.3.2.1)** system-triggered exception (page 60).
	20.2	If the check is not violated, phase is completed.

Process Parameters (SR0080.8+)

The following process parameters define the behavior of the phase.

INSTRUCTION TABLE-SPECIFIC PARAMETERS

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

INSTRUCTION LINK-SPECIFIC PARAMETERS

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 57). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

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

BASIC PARAMETERS

Instruction (SR0080.8.1)

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

CHOICE VALUE PARAMETERS

List of options (SR0080.8.2)

Attribute	Type	Comment
Options	Text (structured)	Defines the available options as key/display text value pairs. Both keys and display texts are unique within a phase.

Option List editor (Framework capability)

The system provides an Option List editor for entering choice items as key/display text value pairs.

One-click completion (SR0080.8.4)

Attribute	Type	Comment
Enabled	Flag	Controls if the phase is automatically completed when an option has been selected.

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Expected value configuration (SR0080.8.5)

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **Expected value key** attribute of the **Expected value definition (SR0080.8.3)** process parameter (page 59) is set.
Display	Flag	Controls if an expected value is displayed during execution. The value is marked as underlined text. Ensure that the **Expected value key** attribute of the **Expected value definition (SR0080.8.3)** process parameter (page 59) is set.
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 **Expected value check (SR0080.3.2.1)** system-triggered exception (page 60).

Expected value definition (SR0080.8.3)

Attribute	Type	Comment
Expected value	String	Defines the expected value.
Default value	String	Defines the pre-selected item in the list of options.

CONFIGURATION OF POST-COMPLETION EXCEPTIONS

Correct value (SR0080.8.6)

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

See also **Correct value (SR0080.3.3.1)** post-completion exception (page 61).

Exceptions (SR0080.3+)

The phase supports user-defined, user-triggered (page 61), system-triggered (page 60), and post-completion exceptions (page 61) and their configuration by means of process parameters (page 55).

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

System-triggered Exceptions (SR0080.3.2+)

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.

Expected value check (SR0080.3.2.1)

Representation of the exception:

- ◼ <Exception text>
  (taken from **Expected value configuration (SR0080.8.5)** process parameter (page 58))
  Expected key/text: <expected key>/<expected text> (taken from **Expected value definition (SR0080.8.3)** process parameter (page 59))
  Actual value: <selected choice value>

    - ◼ Example:
      Expected value check failed.
      Expected key/text: B/Biotech
      Actual key/text: M/Microbiology

Expected value check- Logic (SR0080.3.2.1.1)

- ◼ Trigger: Operator confirms phase

- ◼ Postcondition: Phase is completed

Step	#	Description
Operator confirms phase	10	Phase creates **Expected value check (SR0080.3.2.1)** system-triggered exception.
Operator triggers exception	20	Phase records the exception.
Operator confirms exception	30	Phase is completed.

User-triggered Exceptions

There are no user-triggered exceptions available.

Post-completion Exceptions (SR0080.3.3+)

A post-completion exception is accessible via the Navigator and represented in the list of available post-completion exceptions in the Exception Window, as the description of the exception, and in the batch report.

The following post-completion exceptions are available.

Correct value (SR0080.3.3.1)

The **Correct value** exception allows an operator to correct the selected choice value from the Navigator after the completion of the phase.

> **TIP**
>
> A recorded value could be used within branching. The correction of a value **does not influence** already processed branching decisions.

Representation of the exception:

■ Instruction:
   Select another option. Old value: <Old text>
   Display of list of options according to the **Active mode (SR0080.1.2)** layout
   (page 52)
   Confirm button.

■ <Exception text>
   (taken from **Correct value (SR0080.8.6)** process parameter (page 59))
   Old key/text: <Old key>/<Old text>
   New key/text: <Old key>/<Old text>

   ■ Example:
      Choice corrected (after phase completion).
      Old key/text: Yellow/Yellow appearance of test strip
      New key/text: Blue/Blue appearance of test strip

Correct value - Logic (SR0080.3.3.1.1)

■ Trigger: Phase is completed

■ Postcondition: Post-completion exception is recorded

Step	#	Description
Operator triggers action	10	Phase displays Exception Window.
	20	Operator selects another choice value.
Operator confirms exception	30	Phase checks the value against the settings of the **Expected value definition (SR0080.8.3)** process parameter (page 59).
	30.1	If the limit is violated, only one exception (post-completion exception) is recorded including both, correction and limit violation.

Information Messages

There are no information messages available.

Questions

There are no questions available.

Decisions

There are no decisions available.

Error Messages (SR0080.3.6+)

The following error messages are available.

Phase Configuration-specific Error Messages

The following error messages are available to inform the author in the Phase Preview of Recipe and Workflow Designer about configuration errors of the phase.

In case the error still exists during processing, the phase displays the error message in the **Preview mode (SR0080.1.1)** layout (page 52) and the **Active mode (SR0080.1.2)** layout (page 52). The phase cannot be completed at all.

Invalid expected value configuration (SR0080.3.6.1)

UI text	Comment
Configuration error in expected value definition parameter: expected value must be defined.	This error message is rendered within the Phase Preview of Recipe and Workflow Designer. Message pack: PhaseBaseGetChoiceValue<version> Message ID: ParamValueDefinition_emptyExpectedKey_ErrorMsg

Invalid default value configuration (SR0080.3.6.2)

UI text	Comment
Configuration error in expected value definition parameter: default value must be a list option key.	This error message is rendered within the Phase Preview of Recipe and Workflow Designer. Message pack: PhaseBaseGetChoiceValue<version> Message ID: ParamValueDefinition_emptyDefaultKey_ErrorMsg

Invalid choice item configuration (SR0080.3.6.3)

UI text	Comment
Configuration error in expected value definition parameter: expected value must be a list option key.	This error message is rendered within the Phase Preview of Recipe and Workflow Designer. Message pack: PhaseBaseGetChoiceValue<version> Message ID: ParamValueDefinition_invalidExpectedKey_ErrorMsg

Execution-specific Error Messages

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.

No choice item selected (SR0080.3.6.4)

UI text	Comment
You have to select an option before you can confirm the phase.	Message pack: PhaseBaseGetChoiceValue<version> Message ID: NoChoiceItemSelectedTxt

Output Variables (SR0080.9+)

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.

Option text (SR0080.9.4)

- Data type: String

- Usage: The output variable provides the display text of the selected option.

Option key (SR0080.9.5)

- Data type: String

- Usage: The output variable provides the key value of the selected option.

Upload Image Phase (SR0090+)

The **Upload image** phase allows an operator to upload an image to document processing-related information.

Example use cases are:

- Recording of visual appearance during product test
  During the inspection of a product sample, a picture of the sample can be taken and uploaded.

- Supporting the operator with graphical instructions
  When the operator starts a specific processing step, the phase automatically displays the required illustration.

The phase supports the following file formats: JPG, GIF, and PNG.

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

- In the **Selection** mode, the operator manually selects an image file to be uploaded.

- In the **Loading** mode, the operator triggers the upload of an already pre-defined image file.

- In the **Automatic loading** mode, the phase loads a pre-defined image file automatically.

- In the **Automatic completion** mode, the phase loads a pre-defined image file and is completed automatically without any operator interaction.

The uploaded image and its meta data are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 68). Anomalies that occur during processing are covered by the phase exception handling (page 80) (e.g. required image has not been uploaded).

After completion the phase displays the uploaded image in the Execution Window. The Navigator displays the file name of the uploaded image and provides access to the post-completion exception.
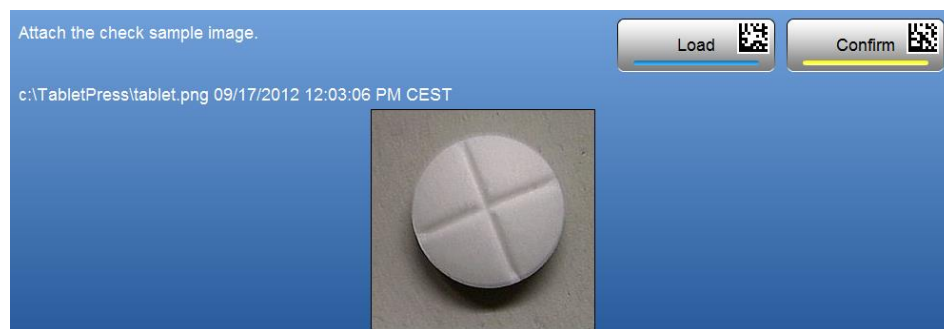
Figure 6: Upload image during execution - Automatic loading mode
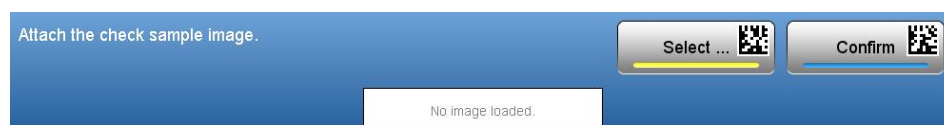
Figure 7: Upload image during execution - Selection mode

Layout

The phase provides individual layouts for its representation during execution (page 66), in the Navigator (page 67), and in the sub-report (page 68).

Representation during Execution (SR0090.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0090.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0090.8.1)** process parameter (page 76))

2. Depends on **Mode (SR0090.8.2)** process parameter (page 76):

   - In **Selection** mode: **Select** button (disabled)

   - In all other modes than **Selection**: **Load** button (disabled)

3. **Confirm** button (disabled).

Active mode (SR0090.1.2)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
   (taken from **Instruction (SR0090.8.1)** process parameter (page 76))

3. Depends on **Mode (SR0090.8.2)** process parameter (page 76):

   ■ In **Selection** mode: **Select** button to open file selection dialog

   ■ In all other modes than **Selection**: **Load** button to load pre-defined image

4. <Full path name of image file>
   <Modification time of image file>
   Uploaded image.

5. **Confirm** button.

Completed mode (SR0090.1.3)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
   (taken from **Instruction (SR0090.8.1)** process parameter (page 76))

3. <Full path name of image file>
   <Modification time of image file>
   Uploaded image.

4. **Confirm** button (completed).

Representation in Navigator (SR0090.4+)

The Navigator provides the following details:

Phase column (Framework capability)

■ <Phase name>

   ■ Example:
     Mixer AB_3X

Information column (SR0090.4.1)

■ Image file name or N/A if no image has been uploaded

   ■ Example: IMG001.JPG

Action column (SR0090.4.2)

■ Replace file, provides exception to replace the uploaded image by a manual file selection.

Representation in Sub-report (SR0090.5+)

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 (SR0090.5.1)

- ◼ Instruction table panel and/or instruction link panel
(only if an instruction table and/or instruction link is defined for the phase)

- ◼ Instruction text

- ◼ Image loaded at: <work center identifier>

- ◼ Image loaded from: <Full path name of image file>

- ◼ Image modified on: <Modification time of image file>

- ◼ Uploaded image

Business Logic (SR0090.2+)

The phase implements the following business logic.

Phase Mode

Business logic related to phase modes.

Selection mode (SR0090.2.1)

- ◼ Function: **Selection** 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 (SR0090.1.2)** layout (page 66).
Operator action	20	The **Select** button opens a file selection dialog, see **Select image (SR0090.2.2)** function (page 71).

Loading mode (SR0090.2.4)

- ■ Function: **Loading (pre-defined file)** 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 (SR0090.1.2)** layout (page 66).
Operator action	20	The **Load** button loads an image file, see **Load image (SR0090.2.5)** function (page 72).

Automatic loading mode (SR0090.2.6)

- ■ Function: **Automatic loading (pre-defined file)** 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 (SR0090.1.2)** layout (page 66).
Phase loads image file	20	Phase loads the image automatically according to the settings of the **File location (SR0090.8.3)** process parameter (page 77).
		If one of the following issues occurs, phase displays an error message:
		1. Image cannot be loaded, **Load image error (SR0090.3.6.1)** error message (page 84).
		2. Image cannot be displayed, **Display image error (SR0090.3.6.2)** error message (page 84).
		3. Image cannot be determined, **Image ambiguous (SR0090.3.6.3)** error message (page 84).
		When the error message has been confirmed, phase returns to the **Active mode (SR0090.1.2)** layout (page 66).
		Otherwise continue with step 40.
Operator interaction	30	In case the automated upload results in an error, the operator can still trigger the upload of an image file manually with the **Load** button, see **Load image (SR0090.2.5)** function (page 72).
		The same checks apply as for step 20.

Step	#	Description
Phase displays image	40	Phase displays the image in the image placeholder of the phase.

Automatic completion mode (SR0090.2.7)

■ Function: **Automatic completion (pre-defined file)** mode of phase

■ Type: Phase mode

■ Trigger: Phase becomes active

■ Postcondition: Phase is completed

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0090.1.2)** layout (page 66).
	20	Phase loads the image automatically according to the settings of the **File location (SR0090.8.3)** process parameter (page 77). If one of the following issues occurs, phase displays an error message: 1. Image cannot be loaded, **Load image error (SR0090.3.6.1)** error message (page 84). 2. Image cannot be displayed, **Display image error (SR0090.3.6.2)** error message (page 84). 3. Image cannot be determined, **Image ambiguous (SR0090.3.6.3)** error message (page 84). When the error message has been confirmed, phase returns to the **Active mode (SR0090.1.2)** layout (page 66). Otherwise continue with step 40.
Operator interaction	30	In case the automated upload results in an error, the operator can still trigger the upload of an image file manually with the **Load** button, see **Load image (SR0090.2.5)** function (page 72). The same checks apply as for step 20. The phase needs to be completed manually by the operator.
Phase displays image	40	Phase displays the image in the image placeholder of the phase.
	50	Phase is completed automatically.

Main Path

Business logic related to the main path:

Select image (SR0090.2.2)

- Function: Image selection and loading

- Type: Main path

- Trigger: Operator opens file selection dialog

- Postcondition: Phase is active

Step	#	Description
Operator opens file selection dialog	10	Phase displays the file selection dialog. Starting directory is taken from the **File location (SR0090.8.3)** process parameter (page 77).
	20	The displayed images are filtered using the configured wildcards (* = multiple characters or ? = single character) for the file name. The file name is taken from the **File location (SR0090.8.3)** process parameter (page 77).
Operator selects an image file	30	Phase closes the file selection dialog.
Phase displays image	40	Phase displays the image in the image placeholder of the phase.

Load image (SR0090.2.5)

■ Function: Image loading

■ Type: Main path

■ Trigger: Operator loads an image file

■ Postcondition: Phase is active

Step	#	Description
Operator loads an image file	10	Phase loads the image according to the settings of the **File location (SR0090.8.3)** process parameter (page 77). If one of the following issues occurs, phase displays an error message: 1. Image cannot be loaded, **Load image error (SR0090.3.6.1)** error message (page 84). 2. Image cannot be displayed, **Display image error (SR0090.3.6.2)** error message (page 84). 3. Image cannot be determined, **Image ambiguous (SR0090.3.6.3)** error message (page 84). When the error message has been confirmed, phase returns to the **Active mode (SR0090.1.2)** layout (page 66). Otherwise continue with step 20.
Phase displays image	20	Phase displays the image in the image placeholder of the phase.

Confirm phase (SR0090.2.3)

- ■ Function: Completion of phase

- ■ Type: Main path

- ■ Trigger: Operator confirms phase

- ■ Postcondition: Phase is completed

Step	#	Description
Operator confirms phase	10	Operator confirms uploaded image.
Phase checks setting of **Mandatory upload check (SR0090.8.6)** process parameter (page 78)	20	If **Mandatory upload check** is enabled: ■ If no image has been uploaded, phase creates the **Mandatory upload check (SR0090.3.2.1)** system-triggered exception (page 80). When the exception has been registered, phase returns to the **Active mode (SR0090.1.2)** layout (page 66). ■ If an image has been uploaded or the system-triggered exception has been registered, continue with step 30.
	30	Phase is completed.

Process Parameters (SR0090.8+)

The following process parameters define the behavior of the phase.

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 75). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

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

BASIC PARAMETERS

Instruction (SR0090.8.1)

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

Mode (SR0090.8.2)

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. **Selection** (default): Operator selects an image file. **Loading (pre-defined file)**: Operator triggers phase to load a pre-defined image file. **Automatic loading (pre-defined file):** Phase automatically loads a pre-defined image file. **Automatic completion (pre-defined file):** Phase automatically loads a pre-defined image file and is completed.

IMAGE PARAMETERS

File location (SR0090.8.3)

Attribute	Type	Comment
Directory path	Text	Defines the directory path of the image file to be loaded. Environment variables are supported (e.g. %USERNAME%). If the **Mode (SR0090.8.2)** process parameter (page 76) is set to **Selection**, the specified path is the starting point for the file selection dialog. The operator can navigate to another directory.
File name	Text	Defines the name of the image file to be loaded. If the **Mode (SR0090.8.2)** process parameter (page 76) is set to **Selection**, all image files are available for selection. If the **Mode (SR0090.8.2)** process parameter (page 76) is **not** set to **Selection**, the file name must be unique even if wildcards are used.

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Mandatory upload check (SR0090.8.6)

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If not, the phase can be completed without an image upload. If so, the phase can only be completed with an image upload or if the **Mandatory upload check (SR00903.2.1)** system-triggered exception (page 80) has been registered.
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. **Mandatory** if the **Enabled** attribute is set to **Yes**.

See also **Mandatory upload check (SR0090.3.2.1)** system-triggered exception (page 80).

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Select manually (SR0090.8.4)

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 **Select manually (SR0090.3.1.1)** user-triggered exception (page 81).

CONFIGURATION OF POST-COMPLETION EXCEPTIONS

Replace file (SR0090.8.5)

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 **Replace file (SR0090.3.3.1)** post-completion exception (page 82).

Exceptions (SR0090.3+)

The phase supports user-defined, user-triggered (page 81), system-triggered (page 80), and post-completion exceptions (page 82) 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 (SR0090.3.2+)

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.

Mandatory upload check (SR0090.3.2.1)

Representation of the exception:

- ■ <Exception text>
  (taken from **Mandatory upload check (SR0090.8.6)** process parameter (page 78))

  - ■ Example:
    Mandatory upload of the sample kit picture is skipped.

Mandatory upload check- Logic (SR0090.3.2.1.1)

- ■ Trigger: Operator confirms phase
- ■ Postcondition: Phase is completed

Step	#	Description
Operator confirms phase	10	Phase creates **Mandatory upload check (SR0090.3.2.1)** system-triggered exception.
Operator triggers exception	20	Phase records the exception.
Operator confirms exception	30	Phase is completed.

User-triggered Exceptions (SR0090.3.1+)

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.

Select manually (SR0090.3.1.1)

The **Select manually** exception allows an operator to select an image file manually. The settings of the **Mode (SR0090.8.2)** process parameter (page 76) are ignored. If an image has already been uploaded, the image is replaced.

Representation during exception handling:

- Instruction:
  To display another image, select its image file.
  Select button.
  <Old image>
  <Full path name of old image file>
  <Modification time of old image file>
  <New image>
  <Full path name of new image file>
  <Modification time of new image file>
  Confirm button.

- Exception text:
  <Exception text>
  (taken from **Select manually (SR0090.8.4)** process parameter (page 79))
  Old image: <Full path name>, <Modification time>
  New image: <Full path name>, <Modification time>

  - Example:
    Manual selection
    Old image: C:\upload\IMG001.JPG, 08/03/2012 10:43:26 AM CEST
    New image: C:\upload\IMG002.JPG, 08/04/2012 10:43:26 AM CEST

Select manually - Logic (SR0090.3.1.1.1)

- Trigger: Exception is selected

- Postcondition: Image is uploaded

Step	#	Description
Operator confirms exception	10	See **Select image (SR0090.2.2)** function (page 71).

Post-completion Exceptions (SR0090.3.3+)

A post-completion exception is accessible via the Navigator and represented in the list of available post-completion exceptions in the Exception Window, as the description of the exception, and in the batch report.

The following post-completion exceptions are available.

Replace file (SR0090.3.3.1)

The **Replace file** exception allows an operator to manually replace the uploaded image from the Navigator after the phase has been confirmed. If an image has already been uploaded, the image is replaced.

> **TIP**
>
> A recorded value could be used within branching. The correction of a value **does not influence** already processed branching decisions.

Representation of the exception:

- Instruction:
  To display another image, select its image file.
  Select button.
  <Old image>
  <Full path name of old image file>
  <Modification time of old image file>
  <New image>
  <Full path name of new image file>
  <Modification time of new image file>
  Confirm button.

■ <Exception text>
(taken from **Replace file (SR0090.8.6)** process parameter (page 79))
Old image: <Full path name>, <Modification time>
New image: <Full path name>, <Modification time>

  ■ Example:
  Image replaced (after phase completion).
  Old image: C:\upload\IMG001.JPG, 08/03/2012 10:43:26 AM CEST
  New image: C:\upload\IMG002.JPG, 08/04/2012 10:43:26 AM CEST

Replace file - Logic (SR0090.3.3.1.1)

■ Trigger: Phase is completed

■ Postcondition: Post-completion exception is recorded

Step	#	Description
Operator triggers action	10	Phase displays Exception Window.
Operator confirms exception	30	See **Select image (SR0090.2.2)** function (page 71). The **Load** button is disabled.

Information Messages

There are no information messages available.

Questions

There are no questions available.

Decisions

There are no decisions available.

Error Messages (SR0090.3.6+)

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.

Load image error (SR0090.3.6.1)

UI text	Comment
Cannot load the image, since the pre-defined file name (invalid file name) does not exist.	Message pack: PhaseBasicUploadImage<version> Message ID: ParamFileLocation_NoFileFound_ErrorMsg
Cannot load the image, since the pre-defined directory (<invalid directory>) does not exist.	Message pack: PhaseBasicUploadImage<version> Message ID: ParamFileLocation_DirDoesNotExist_ErrorMsg
Cannot load the image, since there is no directory path defined.	Message pack: PhaseBasicUploadImage<version> Message ID: ParamFileLocation_NoDirConfigured_ErrorMsg
Cannot load the image, since there is no file name defined.	Message pack: PhaseBasicUploadImage<version> Message ID: ParamFileLocation_NoFileConfigured_ErrorMsg

Display image error (SR0090.3.6.2)

UI text	Comment
Cannot load the image file.	Message pack: PhaseBasicUploadImage<version> Message ID: LoadFile_ErrorMsg

Image ambiguous (SR0090.3.6.3)

UI text	Comment
Cannot load the image, since the pre-defined file location contains more than one matching file.	Message pack: PhaseBasicUploadImage<version> Message ID: ParamFileLocation_FileAmbiguous_ErrorMsg

Output Variables (SR0090.9+)

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.

Image full path (SR0090.9.4)

- Data type: String

- Usage: The output variable provides the full path and file name of the uploaded image.

Image timestamp (SR0090.9.5)

- Data type: Timestamp

- Usage: The output variable provides the modification time of the uploaded image.

Configuration Keys (SR0090.11+)

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

Maximum file size (SR0090.11.1)

- **Phase/UploadImage/uploadMaximumFileSize**
- **Type**: Long
- **Value**: N/A
- **Description**: Defines the maximum allowed file size in bytes of the image file to be uploaded.
  Default: 1000000
- **Range**: <=1000000

Upload PDF Phase (SR0100+)

The **Upload PDF** phase allows an operator to upload a PDF file, to display its content, and to document the content in the batch report.

Example use cases are:

■ Attaching documentation to the batch report
During execution, the operator can upload a PDF file that provides information about test results of a sample.

■ Supporting the operator with instructions that need to be recorded in the batch report
When the operator starts a specific processing step, the phase automatically uploads and displays the required PDF file.

> **TIP**
>
> If the content of a PDF file shall only be displayed, but not included in the batch report, use the **Show URL (SR0120+)** phase (page 109).

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

■ In the **Selection** mode, the operator manually selects a PDF file to be uploaded.

■ In the **Loading** mode, the operator triggers the upload of an already pre-defined PDF file.

■ In the **Automatic loading** mode, the phase loads a pre-defined PDF file automatically.

■ In the **Automatic completion** mode, the phase loads a pre-defined PDF file and is completed automatically without any operator interaction.

The uploaded PDF file and its meta data are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 90). Anomalies that occur during processing are covered by the phase exception handling (page 102) (e.g. required PDF file has not been uploaded).

After completion the phase displays the file name of the uploaded PDF file in the Execution Window. The Navigator displays the file name of the uploaded PDF file and provides access to the post-completion exception.
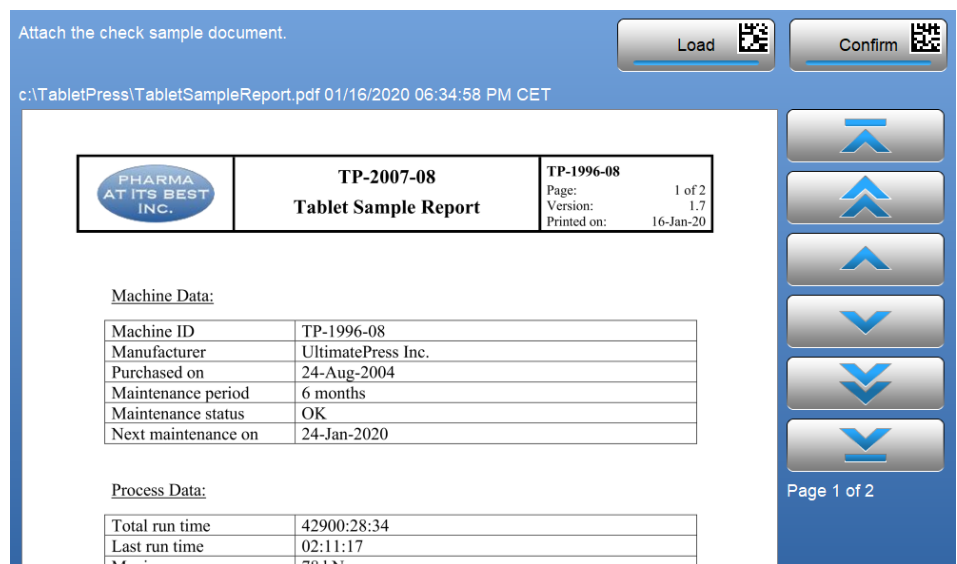
Figure 8: Upload PDF during execution - Automatic loading mode
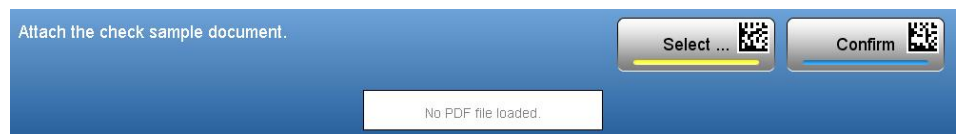
Figure 9: Upload PDF during execution - Selection mode

Layout

The phase provides individual layouts for its representation during execution (page 88), in the Navigator (page 89), and in the sub-report (page 90).

Representation during Execution (SR0100.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0100.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0100.8.1)** process parameter (page 98))

2. Depends on **Mode (SR0100.8.2)** process parameter (page 98):

   ■ In **Selection** mode: **Select** button (disabled)

   ■ In all other modes than **Selection**: **Load** button (disabled)

3. **Confirm** button (disabled).

Active mode (SR0100.1.2)

1. Instruction table panel and/or instruction link panel
(only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
(taken from **Instruction (SR0100.8.1)** process parameter (page 98))

3. Depends on **Mode (SR0100.8.2)** process parameter (page 98):

   ■ In **Selection** mode: **Select** button to open file selection dialog

   ■ In all other modes than **Selection**: **Load** button to load pre-defined PDF file

4. <Full path name of PDF file>
<Modification time of PDF file>
Content of uploaded PDF file.

5. Buttons to navigate through the uploaded PDF file (**First page**, **Previous page**, **Next page**, **Last page**)

6. Page number of the currently displayed page and the total number of pages

7. **Confirm** button.

Completed mode (SR0100.1.3)

1. Instruction table panel and/or instruction link panel
(only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
(taken from **Instruction (SR0100.8.1)** process parameter (page 98))

3. <Full path name of PDF file>
<Modification time of PDF file>
Content of uploaded PDF file.

4. **Confirm** button (completed).

Representation in Navigator (SR0100.4+)

The Navigator provides the following details:

Phase column (Framework capability)

■ <Phase name>

   ■ Example:
SOP: Mixer AB_3X

Information column (SR0100.4.1)

- PDF file name or N/A if no PDF file has been uploaded

    - Example: SOP001.PDF

Action column (SR0100.4.2)

- Replace file, provides exception to replace the uploaded PDF file by a manual file selection.

Representation in Sub-report (SR0100.5+)

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 (SR0100.5.1)

- Instruction table panel and/or instruction link panel
  (only if an instruction table and/or instruction link is defined for the phase)

- Instruction text

- PDF loaded at: \<work center identifier\>

- PDF loaded from: \<Full path name of PDF file\>

- PDF modified on: \<Modification time of PDF file\>

- Content of uploaded PDF file

Business Logic (SR0100.2+)

The phase implements the following business logic.

Phase Mode

Business logic related to phase modes.

Selection mode (SR0100.2.1)

- Function: **Selection** 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 (SR0100.1.2)** layout (page 89).
Operator action	20	The **Select** button opens a file selection dialog, see **Select PDF (SR0100.2.2)** function (page 94).

Loading mode (SR0100.2.4)

- Function: **Loading (pre-defined file)** 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 (SR0100.1.2)** layout (page 89).
Operator action	20	The **Load** button loads a PDF file, see **Load PDF (SR0100.2.5)** function (page 95).

Automatic loading mode (SR0100.2.6)

- Function: **Automatic loading (pre-defined file)** mode of phase

- Type: Phase mode

- Trigger: Phase becomes active

- Postcondition: Phase is active

Step	#	Description
Phase activation	10	Phase displays PDF file
Phase loads PDF file	20	Phase loads the PDF file automatically according to the settings of the **File location (SR0100.8.3)** process parameter (page 99). If one of the following issues occurs, phase displays an error message: 1. PDF cannot be loaded, **Load PDF error (SR0100.3.6.1)** error message (page 105). 2. PDF cannot be displayed, **Display PDF error (SR0100.3.6.2)** error message (page 106). 3. PDF cannot be determined, **PDF ambiguous (SR0100.3.6.3)** error message (page 106). When the error message has been confirmed, phase returns to the **Active mode (SR0100.1.2)** layout (page 89). Otherwise continue with step 40.
Operator interaction	30	In case the automated upload results in an error, the operator can still trigger the upload of a PDF file manually with the **Load** button, see **Load PDF (SR0100.2.5)** function (page 95). The same checks apply as for step 20.
Phase displays PDF file	40	Phase displays the PDF file in the PDF placeholder of the phase.

Automatic completion mode (SR0100.2.7)

■ Function: **Automatic completion (pre-defined file)** mode of phase

■ Type: Phase mode

■ Trigger: Phase becomes active

■ Postcondition: Phase is completed

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0100.1.2)** layout (page 89).
	20	Phase loads the PDF file automatically according to the settings of the **File location (SR0100.8.3)** process parameter (page 99). If one of the following issues occurs, phase displays an error message: 1. PDF cannot be loaded, **Load PDF error (SR0100.3.6.1)** error message (page 105). 2. PDF cannot be displayed, **Display PDF error (SR0100.3.6.2)** error message (page 106). 3. PDF cannot be determined, **PDF ambiguous (SR0100.3.6.3)** error message (page 106). When the error message has been confirmed, phase returns to the **Active mode (SR0100.1.2)** layout (page 89). Otherwise continue with step 40.
Operator interaction	30	In case the automated upload results in an error, the operator can still trigger the upload of a PDF file manually with the **Load** button, see **Load PDF (SR0100.2.5)** function (page 95). The same checks apply as for step 20. The phase needs to be completed manually by the operator.
Phase displays PDF file	40	Phase displays the PDF file in the PDF placeholder of the phase.
	50	Phase is completed automatically.

Main Path

Business logic related to the main path:

Select PDF (SR0100.2.2)

- ■ Function: PDF selection and loading

- ■ Type: Main path

- ■ Trigger: Operator opens file selection dialog

- ■ Postcondition: Phase is active

Step	#	Description
Operator opens file selection dialog	10	Phase displays the file selection dialog. Starting directory is taken from the **File location (SR0100.8.3)** process parameter (page 99).
	20	The displayed PDF files are filtered using the configured wildcards (* = multiple characters or ? = single character) for the file name. The file name is taken from the **File location (SR0100.8.3)** process parameter (page 99).
Operator selects a PDF file	30	Phase closes the file selection dialog.
Phase displays PDF file	40	Phase displays the PDF file in the PDF placeholder of the phase.

Load PDF (SR0100.2.5)

■ Function: PDF loading

■ Type: Main path

■ Trigger: Operator loads a PDF file

■ Postcondition: Phase is active

Step	#	Description
Operator loads a PDF file	10	Phase loads the PDF file according to the settings of the **File location (SR0100.8.3)** process parameter (page 99). If one of the following issues occurs, phase displays an error message: 1. PDF cannot be loaded, **Load PDF error (SR0100.3.6.1)** error message (page 105). 2. PDF cannot be displayed, **Display PDF error (SR0100.3.6.2)** error message (page 106). 3. PDF cannot be determined, **PDF ambiguous (SR0100.3.6.3)** error message (page 106). When the error message has been confirmed, phase returns to the **Active mode (SR0100.1.2)** layout (page 89). Otherwise continue with step 20.
Phase displays PDF file	20	Phase displays the PDF file in the PDF placeholder of the phase.

Confirm phase (SR0100.2.3)

- Function: Completion of phase

- Type: Main path

- Trigger: Operator confirms phase

- Postcondition: Phase is completed

Step	#	Description
Operator confirms phase	10	Operator confirms uploaded PDF file.
Phase checks setting of **Mandatory upload check (SR0100.8.6)** process parameter (page 100)	20	If **Mandatory upload check** is enabled: - If no PDF file has been uploaded, phase creates the **Mandatory upload check (SR0100.3.2.1)** system-triggered exception (page 102). When the exception has been registered, phase returns to the **Active mode (SR0100.1.2)** layout (page 89). - If a PDF file has been uploaded or the system-triggered exception has been registered, continue with step 30.
	30	Phase is completed.

Process Parameters (SR0100.8+)

The following process parameters define the behavior of the phase.

INSTRUCTION TABLE-SPECIFIC PARAMETERS

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

INSTRUCTION LINK-SPECIFIC PARAMETERS

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 97). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.

Attribute	Type	Comment
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

BASIC PARAMETERS

Instruction (SR0100.8.1)

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

Mode (SR0100.8.2)

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. **Selection** (default): Operator selects a PDF file. **Loading (pre-defined file)**: Operator triggers phase to load a pre-defined PDF file. **Automatic loading (pre-defined file)**: Phase automatically loads a pre-defined PDF file. **Automatic completion (pre-defined file)**: Phase automatically loads a pre-defined PDF file and is completed.

PDF FILE PARAMETERS

File location (SR0100.8.3)

Attribute	Type	Comment
Directory path	Text	Defines the directory path of the PDF file to be loaded. Environment variables are supported (e.g. %USERNAME%). If the **Mode (SR0100.8.2)** process parameter (page 98) is set to **Selection**, the specified path is the starting point for the file selection dialog. The operator can navigate to another directory.
File name	Text	Defines the name of the PDF file to be loaded. If the **Mode (SR0100.8.2)** process parameter (page 98) is set to **Selection**, wildcards are supported to restrict the number of displayed PDF files. Example: rep*.pdf displays all PDF files starting with rep. If the **Mode (SR0100.8.2)** process parameter (page 98) is **not** set to **Selection**, the file name must be unique even if wildcards are used.

Mandatory upload check (SR0100.8.6)

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If not, the phase can be completed without a PDF file upload. If so, the phase can only be completed with a PDF file or if the **Mandatory upload check (SR01003.2.1)** system-triggered exception (page 102) has been registered.
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. **Mandatory** if the **Enabled** attribute is set to **Yes**.

See also **Mandatory upload check (SR0100.3.2.1)** system-triggered exception (page 102).

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Select manually (SR0100.8.4)

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 **Select manually (SR0100.3.1.1)** user-triggered exception (page).

CONFIGURATION OF POST-COMPLETION EXCEPTIONS

Replace file (SR0100.8.5)

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 **Replace file (SR0100.3.3.1)** post-completion exception (page).

Exceptions (SR0100.3+)

The phase supports user-defined, user-triggered (page 103), system-triggered (page 102), and post-completion exceptions (page 104) and their configuration by means of process parameters (page 96).

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

System-triggered Exceptions (SR0100.3.2+)

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.

Mandatory upload check (SR0100.3.2.1)

Representation of the exception:

■ <Exception text>
(taken from **Mandatory upload check (SR0100.8.6)** process parameter (page 100))

■ Example:
Mandatory upload of the Sample Report is skipped.

Mandatory upload check- Logic (SR0100.3.2.1.1)

■ Trigger: Operator confirms phase

■ Postcondition: Phase is completed

Step	#	Description
Operator confirms phase	10	Phase creates **Mandatory upload check (SR0100.3.2.1)** system-triggered exception.
Operator triggers exception	20	Phase records the exception.
Operator confirms exception	30	Phase returns to the **Active mode (SR0100.1.2)** layout (page 89).

User-triggered Exceptions (SR0100.3.1+)

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.

Select manually (SR0100.3.1.1)

The **Select manually** exception allows an operator to select a PDF file manually. The settings of the **Mode (SR0100.8.2)** process parameter (page 98) are ignored. If a PDF file has already been uploaded, the PDF file is replaced.

Representation during exception handling:

■ Instruction:
To display another document, select its PDF file.
Select button.
\<Old PDF file\>
\<Full path name of old PDF file\>
\<Modification time of old PDF file\>
\<New PDF\>
\<Full path name of new PDF file\>
\<Modification time of new PDF file\>
Confirm button.

■ Exception text:
\<Exception text\>
(taken from **Select manually (SR0100.8.4)** process parameter (page 101))
Old PDF file: \<Full path name\>, \<Modification time\>
New PDF file: \<Full path name\>, \<Modification time\>

   ■ Example:
Manual selection
Old PDF file: C:\upload\SOP001.PDF, 08/03/2012 10:43:26 AM CEST
New PDF file: C:\upload\SOP002.PDF, 08/04/2012 10:43:26 AM CEST

Select manually - Logic (SR0100.3.1.1.1)

■ Trigger: Exception is selected

■ Postcondition: PDF file is uploaded

Step	#	Description
Operator confirms exception	10	See **Select PDF (SR0100.2.2)** function (page 94).

Post-completion Exceptions (SR0100.3.3+)

A post-completion exception is accessible via the Navigator and represented in the list of available post-completion exceptions in the Exception Window, as the description of the exception, and in the batch report.

The following post-completion exceptions are available.

Replace file (SR0100.3.3.1)

The **Replace file** exception allows an operator to manually replace the uploaded PDF file from the Navigator after the phase has been confirmed. If a PDF file has already been uploaded, the PDF file is replaced.

> **TIP**
>
> A recorded value could be used within branching. The correction of a value **does not influence** already processed branching decisions.

Representation of the exception:

- Instruction:
  To display another document, select its PDF file.
  Select button.
  <Old PDF file>
  <Full path name of old PDF file>
  <Modification time of old PDF file>
  <New PDF>
  <Full path name of new PDF file>
  <Modification time of new PDF file>
  Confirm button.

- <Exception text>
  (taken from **Replace file (SR0100.8.6)** process parameter (page 101))
  Old PDF file: <Full path name>, <Modification time>
  New PDF file: <Full path name>, <Modification time>

  - Example:
    PDF file replaced (after phase completion).
    Old PDF file: C:\upload\SOP001.PDF, 08/03/2012 10:43:26 AM CEST
    New PDF file: C:\upload\SOP002.PDF, 08/04/2012 10:43:26 AM CEST

Replace file - Logic (SR0100.3.3.1.1)

- Trigger: Phase is completed

- Postcondition: Post-completion exception is recorded

Step	#	Description
Operator triggers action	10	Phase displays Exception Window.
Operator confirms exception	30	See **Select PDF (SR0100.2.2)** function (page 94). The **Load** button is disabled.

Information Messages

There are no information messages available.

Questions

There are no questions available.

Decisions

There are no decisions available.

Error Messages (SR0100.3.6+)

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.

Load PDF error (SR0100.3.6.1)

UI text	Comment
Cannot load the PDF file, since the pre-defined file name (invalid file name) does not exist.	Message pack: PhaseBasicUploadPDF<version> Message ID: ParamFileLocation_NoFileFound_ErrorMsg
Cannot load the PDF file, since the pre-defined directory (<invalid directory>) does not exist.	Message pack: PhaseBasicUploadPDF<version> Message ID: ParamFileLocation_DirDoesNotExist_ErrorMsg

UI text	Comment
Cannot load the PDF file, since there is no directory path defined.	Message pack: PhaseBasicUploadPDF<version> Message ID: ParamFileLocation_NoDirConfigured_ErrorMsg
Cannot load the PDF file, since there is no file name defined.	Message pack: PhaseBasicUploadPDF<version> Message ID: ParamFileLocation_NoFileConfigured_ErrorMsg
Cannot display the PDF file, since it is password-protected.	Message pack: PhaseBasicUploadPDF<version> Message ID: PdfPasswordProtected_ErrorMsg
Cannot upload the PDF file, since it is encrypted and does not allow printing.	Message pack: PhaseBasicUploadPDF<version> Message ID: EncryptedFile_PrintingNotAllowed_ErrorMsg

Display PDF error (SR0100.3.6.2)

UI text	Comment
Cannot display the PDF file.	Message pack: PhaseBasicUploadPDF<version> Message ID: displayPDF_ErrorMsg

PDF ambiguous (SR0100.3.6.3)

UI text	Comment
Cannot load the PDF file, since the pre-defined file location contains more than one matching file.	Message pack: PhaseBasicUploadPDF<version> Message ID: ParamFileLocation_FileAmbiguous_ErrorMsg

Output Variables (SR0100.9+)

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.

PDF full path (SR0100.9.1)

- Data type: String

- Usage: The output variable provides the full path and file name of the uploaded PDF file.

PDF timestamp (SR0100.9.2)

- Data type: Timestamp

- Usage: The output variable provides the modification time of the uploaded PDF file.

Configuration Keys (SR0100.11+)

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

Maximum file size (SR0100.11.1)

- **Phase/UploadPdf/uploadMaximumFileSize**
- **Type**: Long
- **Value**: N/A
- **Description**: Defines the maximum allowed file size in bytes of the PDF file to be uploaded.
  Default: 5000000
- **Range**: <=5000000

Show URL Phase (SR0120+)

The **Show URL** phase allows to display PDF documents accessible via a URL.

> **TIP**
>
> Due to the diversity of available user authentication and SSL security solutions, the **Show URL** phase does not support user authentication or SSL security by default. However, the phase can be extended by a system integrator in order to support required customer-specific solutions.

An example use case is:

■ Display an SOP
An SOP is stored on the intranet or the file system. This very SOP can be displayed to the operator during execution.

The URL and the instruction text are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 111).

Anomalies that occur during processing are covered by the phase exception handling (page 116) (e.g. document could not be loaded).

The URL of the displayed document is shown on the detail information button in the Navigator (e.g. www.PharmaAtItsBest.sop1.pdf).
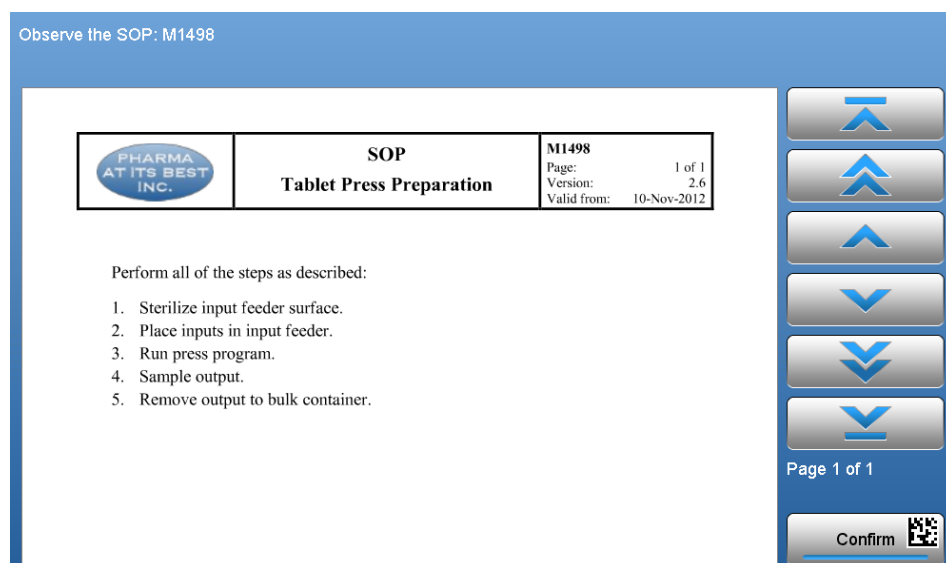
Figure 10: Show URL during execution

Layout

The phase provides individual layouts for its representation during execution (page 110), in the Navigator (page 111), and in the sub-report (page 111).

Representation during Execution (SR0120.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0120.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0120.8.1)** process parameter (page 114))

2. <URL>
   (taken from **Document (SR0120.8.2)** process parameter (page 114))

3. **Confirm** button (disabled).

Active mode (SR0120.1.2)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <URL>
   (taken from **Document (SR0120.8.2)** process parameter (page 114))

3. <Instruction text>
   (taken from **Instruction (SR0120.8.1)** process parameter (page 114))

4. <Document>, in a PDF viewer

5. Buttons to navigate through the document (**First page**, **Previous page**, **Previous line**, **Next line**, **Next page**, **Last page**)

6. **Confirm** button.

Completed mode (SR0120.1.3)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
   (taken from **Instruction (SR0120.8.1)** process parameter (page 114))

3. <URL>
   (taken from **Document (SR0120.8.2)** process parameter (page 114))

4. **Confirm** button (completed).

Representation in Navigator (SR0120.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- ■ <Phase name>
    - ■ Example:
      Mix inputs

Information column (SR0120.4.1)

- ■ <Phase name>
    - ■ Example: Cleaning SOP V1.0

Action column (SR0120.4.2)

- ■ There are no phase-specific actions available.

Representation in Sub-report (SR0120.5+)

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 (SR0120.5.1)

- ■ Instruction table panel and/or instruction link panel
  (only if an instruction table and/or instruction link is defined for the phase)
- ■ URL
- ■ Instruction text

Business Logic (SR0120.2+)

The phase implements the following business logic.

Display document (SR0120.2.1)

- Function: Display a document

- Trigger: Phase becomes active

- Postcondition: Document was displayed

Step	#	Description
Phase activation	10	Phase displays document.

Process Parameters (SR0120.8+)

The following process parameters define the behavior of the phase.

INSTRUCTION TABLE-SPECIFIC PARAMETERS

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

INSTRUCTION LINK-SPECIFIC PARAMETERS

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 113). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

Attribute	Type	Comment
Link text	Text	Text to be used as link. For any text enclosed in curly brackets within the instruction text you can define a link with the **Link URL** attribute. Including the brackets in the link text is optional. Maximum length is 80 characters.

Attribute	Type	Comment
Link URL	Text	URL of the file to be displayed. The link opens the external application assigned to the file type by the operating system. Maximum length is 256 characters.

BASIC PARAMETERS

Instruction (SR0120.8.1)

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

Document (SR0120.8.2)

Attribute	Type	Comment
URL	Text	URL of the document to be displayed. The document will be shown within a PDF viewer.

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Loading failed (SR0120.8.3)

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 **Loading failed (SR0120.3.2.1)** system-triggered exception (page 116).

Exceptions (SR0120.3+)

The phase supports user-defined, user-triggered (page 117), system-triggered (page 116), and post-completion exceptions (page 117) and their configuration by means of process parameters (page 112).

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

System-triggered Exceptions (SR0120.3.2+)

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.

Loading failed (SR0120.3.2.1)

Representation of the exception:

In case the file cannot be loaded e.g. because the URL is not valid or the file is password-protected, the system triggers this exception.

- ■ <Exception text>
  (taken from **Loading failed (SR0120.8.3)** process parameter (page 115))
  Cannot load <URL>

  - ■ Example:
    SOP 0001 could not be loaded.
    Cannot load www.PharmaAtItsBest.sop0001.pdf

Loading failed - Logic (SR0120.3.2.1.1)

- ■ Trigger: Loading of document failed

- ■ Postcondition: Exception is recorded

Step	#	Description
Operator triggers exception	10	Phase records the exception.

User-triggered Exceptions

There are no user-triggered exceptions available.

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

There are no error messages available.

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.

Create Workflow Phase (SR0130+)

The **Create workflow** phase allows to create a workflow during the execution of an order or workflow.
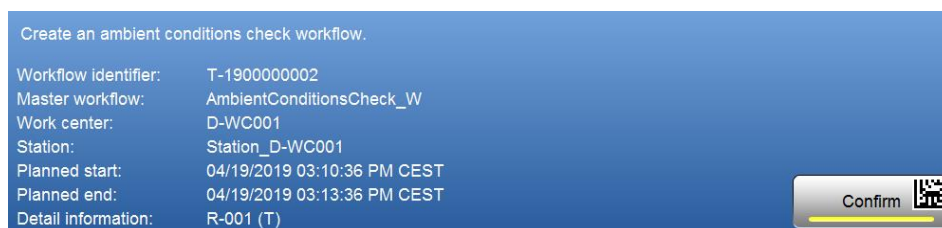
Example use cases are:

- During order execution the container to be used is contaminated and needs to be cleaned. An ad-hoc workflow to clean the container is created and appended to the order.

- The flow rate of a filter is too low and the filter needs to be rinsed. An ad-hoc workflow is created to rinse the filter.

The instruction text, the used master workflow identifier and the workflow identifier of the created workflow, the optional configured work center, station, planned start, planned end, and detail information are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 123).

After completion the phase displays the defined data.

The navigator displays the identifier of the created workflow.

Create an ambient conditions check workflow.	
Workflow identifier:	T-1900000002
Master workflow:	AmbientConditionsCheck_W
Work center:	D-WC001
Station:	Station_D-WC001
Planned start:	04/19/2019 03:10:36 PM CEST
Planned end:	04/19/2019 03:13:36 PM CEST
Detail information:	R-001 (T)

Figure 11: Create workflow during execution

Layout

The phase provides individual layouts for its representation during execution (page 120), in the Navigator (page 122), and in the sub-report (page 123).

Representation during Execution (SR0130.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0130.1.1)

1. \<Instruction text\>
    (taken from **Instruction (SR0130.8.1)** process parameter (page 130))

2. Workflow identifier (taken from **Workflow definition (SR0130.8.3)** process parameter (page 130))
    Master workflow (taken from **Master workflow (SR0130.8.4)** process parameter (page 130))
    Work center (taken from **Work center (SR0130.8.6)** process parameter (page 131))
    Station (taken from **Station (SR0130.8.7)** process parameter (page 131))
    Planned start (taken from **Planned start (SR0130.8.9)** process parameter (page 132))
    Planned end (taken from **Planned end (SR0130.8.10)** process parameter (page 132))
    Detail information (taken from **Detail information (SR0130.8.11)** process parameter (page 132))
    The system displays only the labels and their data for which input has been configured and is already evaluated.

3. **Confirm** button (disabled).

Active mode (SR0130.1.2)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
   (taken from **Instruction (SR0130.8.1)** process parameter (page 130))

3. Workflow identifier (taken from **Workflow definition (SR0130.8.3)** process parameter (page 130))
   Master workflow (taken from **Master workflow (SR0130.8.4)** process parameter (page 130))
   Work center (taken from **Work center (SR0130.8.6)** process parameter (page 131))
   Station (taken from **Station (SR0130.8.7)** process parameter (page 131))
   Planned start (taken from **Planned start (SR0130.8.9)** process parameter (page 132))
   Planned end (taken from **Planned end (SR0130.8.10)** process parameter (page 132))
   Detail information (taken from **Detail information (SR0130.8.11)** process parameter (page 132))
   The system displays only the labels and their data for which input has been configured and is already evaluated.

4. **Confirm** button.

Completed mode (SR0130.1.3)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
   (taken from **Instruction (SR0130.8.1)** process parameter (page 130))

3. Workflow identifier (taken from **Workflow definition (SR0130.8.3)** process parameter (page 130)) or the generated workflow identifier if the parameter is not specified)

   Master workflow (taken from **Master workflow (SR0130.8.4)** process parameter (page 130))

   Work center (taken from **Work center (SR0130.8.6)** process parameter (page 131))

   Station (taken from **Station (SR0130.8.7)** process parameter (page 131))

   Planned start (taken from **Planned start (SR0130.8.9)** process parameter (page 132))

   Planned end (taken from **Planned end (SR0130.8.10)** process parameter (page 132))

   Detail information (taken from **Detail information (SR0130.8.11)** process parameter (page 132))

   The system displays only the labels and their data for which input has been configured or a default was generated.

4. **Confirm** button (completed).

Representation in Navigator (SR0130.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- ■ <Phase name>

  - ■ Example:
    Create cleaning workflow

Information column (SR0130.4.1)

- ■ <Workflow identifier>

  - ■ Example: WF1900002485

Action column

- ■ There are no phase-specific actions available.

Representation in Sub-report (SR0130.5+)

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 (SR0130.5.1)

- ■ Instruction table panel and/or instruction link panel
  (only if an instruction table and/or instruction link is defined for the phase)

- ■ Instruction text

- ■ The system only prints the labels and their data for which input has been configured or a default was generated:

    - ■ Workflow identifier (taken from **Workflow definition (SR0130.8.3)** process parameter (page 130)) or the generated default value

    - ■ Master workflow (taken from **Master workflow (SR0130.8.4)** process parameter (page 130))

    - ■ Work center (taken from **Work center (SR0130.8.6)** process parameter (page 131))

    - ■ Station (taken from **Station (SR0130.8.7)** process parameter (page 131))

    - ■ Planned start (taken from **Planned start (SR0130.8.9)** process parameter (page 132))

    - ■ Planned end (taken from **Planned end (SR0130.8.10)** process parameter (page 132))

    - ■ Detail information (taken from **Detail information (SR0130.8.11)** process parameter (page 132))

Business Logic (SR0130.2+)

The phase implements the following business logic.

Phase Mode

Business logic related to phase modes.

Manual completion mode (SR0130.2.1)

- 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 (SR0130.1.2)** layout (page 121).
Phase completion	30	See **Confirm phase (SR0130.2.3)** function (page 125).

Automatic completion mode (SR0130.2.2)

- Function: **Automatic completion** mode of phase

- Type: Phase mode

- Trigger: Phase becomes active

- Postcondition: Phase is completed

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0130.1.2)** layout (page 121).
	20	The system tries to confirm the phase immediately as if the operator had used the **Confirm** button of the phase to trigger the **Confirm phase (SR0130.2.3)** function (page 125).

Main Path

Business logic related to the main path:

Confirm phase (SR0130.2.3)

- ■ Function: Completion of phase

- ■ Trigger: Operator confirms phase

- ■ Postcondition: Phase is completed

Step	#	Description
Operator confirms phase	10	Operator confirms that the workflow can be created.
System checks if work center and station exist	20	If the work center configured with the **Work center (SR0130.8.6)** process parameter (page 131) does not exist, phase displays the **Work center does not exist (SR0130.3.6.3)** error message (page 136). If the station configured with the **Station (SR0130.8.7)** process parameter (page 131) does not exist, phase displays the **Station does not exist (SR0130.3.6.4)** error message (page 136). If both the configured work center and the configured station do not exist, a combined error message is displayed. When the error message has been confirmed, phase returns to the **Active mode (SR0130.1.2)** layout (page 121). Otherwise continue with step 30.
System checks consistency between work center and station	30	If the value of the **Work center (SR0130.8.6)** process parameter (page 131) is not the work center of the station configured with the **Station (SR0130.8.7)** process parameter (page 131), phase displays the **Station and work center inconsistency (SR0130.3.6.1)** error message (page 135). When the error message has been confirmed, phase returns to the **Active mode (SR0130.1.2)** layout (page 121). Otherwise continue with step 35.
System checks master workflow	35	If no approved master workflow recipe exists for the identifier configured with the **Master workflow (SR0130.8.6)** process parameter (page 130), phase displays the **Master workflow not approved (SR0130.3.6.2)** error message (page 135). When the error message has been confirmed, phase returns to the **Active mode (SR0130.1.2)** layout (page 121). Otherwise continue with step 40.

Step	#	Description
System creates workflow	40	If a workflow with the identifier configured with the **Workflow definition (SR0130.8.6)** process parameter (page 130) already exists, phase displays the **Duplicate workflow identifier (SR0130.3.6.5)** error message (page 136). When the error message has been confirmed, phase returns to the **Active mode (SR0130.1.2)** layout (page 121). Otherwise the workflow with the configured data for - planned start according to the **Planned start (SR0130.8.9)** process parameter (page 132) - planned end according to the **Planned end (SR0130.8.10)** process parameter (page 132) - detail information according to the **Detail information (SR0130.8.11)** process parameter (page 132) - work center and station according to the Work center and station determination (SR0130.2.4) function (page 127) is created and released. If the phase is running in the context of an order unit procedure, a treatment ID is defined for the order, and the system is configured accordingly (see **Enable and Configure Treatment ID Support (SR0130.11.1)** configuration key) (page 138), the workflow inherits the treatment ID from the order.
System appends workflow	45	If the **Append Workflow (SR0130.8.5)** process parameter (page 131) is set to **Yes** and the phase is running in the context of an order unit procedure, the created workflow is appended to the unit procedure. At start of the workflow the system will not ask the user if the workflow shall be appended to an order.
System starts workflow	50	If the **Start workflow automatically (SR0130.8.8)** process parameter (page 131) is set to **Yes** and - a station is set with the **Station (SR0130.8.7)** process parameter (page 131), then the workflow is started at this station, - the workflow is dispatched to one work center, then the workflow is started at this work center, - the workflow is dispatched to several work centers, then the workflow is only started at the current work center and only if it is one of the work centers to which the workflow has been dispatched, otherwise the workflow is not started, - the workflow is dispatched neither to a station nor to a work center, then the workflow is started at the current work center. Starting a workflow means it will be available in the Cockpit with its unit procedure and startable operation(s).
	60	Phase is completed.

Work center and station determination (SR0130.2.4)

■ Function: Determine the work center and station for the workflow

■ Trigger: Phase creates the workflow

■ Postcondition: Workflow is dispatched to work center and station

Step	#	Description
Station defined	10	If the **Station (SR0130.8.7)** process parameter (page 130) has a value, the workflow unit procedure will be dispatched to the station and the work center of the station. Station assignments defined with the master workflow are ignored. Otherwise continue with step 20.
Work center defined	20	If the **Work center (SR0130.8.6)** process parameter (page 131) has a value, the unit procedure of the workflow will be dispatched to this work center. Station assignments defined with the master workflow are ignored. Otherwise continue with step 30.
Stations defined at recipe	30	If the master workflow has stations assigned to its unit procedure, the unit procedure of the workflow will be dispatched to the stations and work centers defined with the master workflow. Otherwise continue with step 40.
Work center defined at recipe	40	If the master workflow has only work centers assigned to its unit procedure, the unit procedure of the workflow will be dispatched to these work centers.

Process Parameters (SR0130.8+)

The following process parameters define the behavior of the phase.

INSTRUCTION TABLE-SPECIFIC PARAMETERS

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 129). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

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

BASIC PARAMETERS

Instruction (SR0130.8.1)

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

Mode (SR0130.8.2)

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. **Manual completion** (default): Operator confirms the phase. **Automatic completion**: Phase automatically tries to confirm the phase.

Workflow definition (SR0130.8.3)

Attribute	Type	Comment
Identifier	String	Optional parameter to define a unique workflow identifier for the workflow to be created. If empty, a workflow identifier with the configured default prefix is generated. Maximum length is 12 characters.

Master workflow (SR0130.8.4)

Attribute	Type	Comment
Identifier	String	Defines the master workflow identifier without version to be used for generating the workflow. The Master Workflow Selection editor (**Framework capability**) only lists **Scheduled** and **Approved** master workflows. Maximum length is 50 characters.

Append workflow (SR0130.8.5)

Attribute	Type	Comment
Enabled	Flag	Controls if the created workflow is automatically appended to the current unit procedure. If the phase runs in the context of a workflow, the parameter is ignored. Default setting: **Yes**

Work center (SR0130.8.6)

Attribute	Type	Comment
Identifier	String	Optional parameter to define the work center to which the created workflow is dispatched. The Work Center Selection editor is a **Framework capability.** Maximum length is 20 characters.

Station (SR0130.8.7)

Attribute	Type	Comment
Identifier	String	Optional parameter to define the station to which the created workflow is dispatched. The Station Selection editor is a **Framework capability.** Maximum length is 20 characters.

Start workflow automatically (SR0130.8.8)

Attribute	Type	Comment
Enabled	Flag	Controls if the created workflow will be available in the cockpit if it is dispatched to a work center or a station. Default setting: **No**

Planned start (SR0130.8.9)

Attribute	Type	Comment
Timestamp	Timestamp	Optional information shown to an operator in the workflow processing list of the work center to which a workflow hat been dispatched, unless the workflow is started automatically.

Planned end (SR0130.8.10)

Attribute	Type	Comment
Timestamp	Timestamp	Optional information shown to an operator in the workflow processing list of the work center to which a workflow hat been dispatched, unless the workflow is started automatically.

Detail information (SR0130.8.11)

Attribute	Type	Comment
Value	String	Optional information shown to an operator in the workflow processing list of the work center to which a workflow hat been dispatched, unless the workflow is started automatically. Maximum length is 20 characters.

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Skip workflow creation (SR0130.8.12)

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.

Exceptions (SR0130.3+)

The phase supports user-defined, user-triggered (page 81), system-triggered (page 80), and post-completion exceptions (page 82) 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

There are no system-triggered exceptions available.

User-triggered Exceptions (SR0130.3.1+)

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.

Skip workflow creation (SR0130.3.1.1)

The **Skip workflow creation** exception allows an operator to skip the phase if the situation on the shop floor does not require the creation of a workflow anymore. To solve issues caused by phase misconfiguration or error cases, the framework provides the **Abort Phase (SR1200.3.1)** and the **Repair Phase (SR1200.3.2)** functions, see "Functional Requirement Specification Execution Framework" [A1] (page 179).

Representation during exception handling:

- Instruction
  Skip the workflow creation.
  Confirm button.

- Exception text:
  <Exception text>
  (taken from **Skip workflow parameter (SR0130.8.12)** process parameter (page 133))
  Skip the workflow creation.

  - Example:
    Supervisor needs to be informed to create the required workflow!
    Skip the workflow creation.

Skip workflow creation - Logic (SR0130.3.1.1.1)

- Trigger: Exception is selected

- Postcondition: **Create workflow** phase is skipped.

Step	#	Description
Operator confirms exception	10	Phase shows exception description to be signed according to **Skip workflow creation (SR0130.8.12)** process parameter (page 133).
Operator signs exception	20	- Phase records the exception. - Operator completes the phase manually without creation of a workflow.

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 (SR0130.3.6+)

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.

Station and work center inconsistency (SR0130.3.6.1)

UI text	Comment
\<Information that the configured work center is not the work center of the configured station.\>	Message pack: PhaseEbrCreateWorkflow\<version\> Message ID: WorkcenterStationInconsistent_Error

Master workflow not approved (SR0130.3.6.2)

UI text	Comment
\<Information that there is no approved master workflow with the configured identifier available.\>	Message pack: PhaseEbrCreateWorkflow\<version\> Message ID: NoValidWorkflow_Error

Work center does not exist (SR0130.3.6.3)

UI text	Comment
<Information that the configured work center does not exist.>	Message pack: PhaseEbrCreateWorkflow<version> Message ID: NoValidWorkcenter_Error

Station does not exist (SR0130.3.6.4)

UI text	Comment
<Information that the configured station does not exist.>	Message pack: PhaseEbrCreateWorkflow<version> Message ID: NoValidStation_Error

Duplicate workflow identifier (SR0130.3.6.5)

UI text	Comment
<Information that the configured workflow identifier is already in use.>	Message pack: PhaseEbrCreateWorkflow<version> Message ID: BadWorkflowIdentifier_Error

Output Variables (SR0130.9+)

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.

Workflow identifier (SR0130.9.1)

■ Data type: String

■ Usage: The output variable provides the identifier of the created workflow taken from the **Workflow definition (SR0130.8.3)** process parameter (page 130) or the generated default value.

Planned start (SR0130.9.2)

■ Data type: Timestamp

■ Usage: The output variable provides the planned start time of the workflow taken from **Planned start (SR0130.8.9)** process parameter (page 132).

Planned end (SR0130.9.3)

■ Data type: Timestamp

■ Usage: The output variable provides the planned end time of the workflow taken from **Planned end (SR0130.8.10)** process parameter (page 132).

Detail information (SR0130.9.4)

■ Data type: String

■ Usage: The output variable provides the detail information about the workflow taken from **Detail information (SR0130.8.11)** process parameter (page 132).

Creation result (SR0130.9.5)

- Data type: String

- Values: CREATED, SKIPPED

- Usage: The output variable states if a workflow was created or the phase was skipped.

  - The value is CREATED if the creation of the workflow was successful.

  - The value is SKIPPED if no workflow was created and the phase was skipped by the **Skip workflow creation (SR0130.3.1.1)** user triggered exception (page).

Configuration Keys (SR0130.11+)

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

Enable and Configure Treatment ID Support (SR0130.11.1)

- **Phase/TreatmentIDSupport**

- **Type**: String

- **Value**: Off

- **Description**: The configuration applies to material identification with the **D Identify material** phase and the **Identify material** phase and to workflow creation with the **Create workflow** phase.
  If the value is set to **Off**, no treatment ID is checked during material identification and a created workflow does not inherit the treatment ID of the order.
  If the value is set to **TreatmentIDMandatory**, all orders need to have a treatment ID set. If the identified batch has a treatment ID, it must be equal to the order's treatment ID.
  If the value is set to **TreatmentIDOptional**, an order does not have to have a treatment ID set. If the identified batch has a treatment ID, it has to be equal to the order's treatment ID.

- **Evaluated**: At sublot or batch identification and at workflow creation.

- **Range**: [Off, TreatmentIDMandatory, TreatmentIDOptional]

Write Context Data Phase (SR0140+)

The **Write context data** phase allows to store processing data from an order or workflow for later use.

Example use cases are:

■ Store the equipment identifier of a filter to allow a filter test workflow to check that it runs the test against the planned filter.

■ Store and increase a counter value in a loop for a number of used containers until a planned number of containers per type has been filled.

Each recorded value is stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 144).

Anomalies that occur during processing are covered by the phase exception handling (page 158) (e.g. context and key check).

After completion, the phase displays the data written to the context.

The Navigator displays the identifier of the context.

View and confirm the context data.	
Context ID: T-1900000002	
Key	**Value**
EQ-ID	R-001 (T)
ID-TIME	4/19/2019 1:36:55 PM CEST
	Confirm

Figure 12: Write context data during execution

Layout

The phase provides individual layouts for its representation during execution (page 140), in the Navigator (page 144), and in the sub-report (page 144).

Representation during Execution (SR0140.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0140.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0140.8.1)** process parameter (page 153))

2. Context identifier (taken from **Context definition (SR0140.8.3)** process parameter (page 153).

3. **Confirm** button (disabled).

Active mode (SR0140.1.2)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
   (taken from **Instruction (SR0140.8.1)** process parameter (page 153))

3. Context identifier (taken from **Context definition (SR0140.8.3)** process parameter (page 153).

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

   ■ **BigDecimal Value Bundle**:

      ■ Key
        (taken from the [BigDecimal] **Master bundle identifier (SR0140.8.6)** process parameter (page 155))

■ Value
(taken from the [BigDecimal] **Master bundle identifier (SR0140.8.6)**
process parameter (page 155))

- **Boolean Value Bundle**:

  ■ Key
  (taken from the [Boolean] **Master bundle identifier (SR0140.8.7)**
  process parameter (page 155))

  ■ Value
  (taken from the [Boolean] **Master bundle identifier (SR0140.8.7)**
  process parameter (page 155))

- **Duration Value Bundle**:

  ■ Key
  (taken from the [Duration] **Master bundle identifier (SR0140.8.8)**
  process parameter (page 156))

  ■ Value
  (taken from the [Duration] **Master bundle identifier (SR0140.8.8)**
  process parameter (page 156))

- **Long Value Bundle**:

  ■ Key
  (taken from the [Long] **Master bundle identifier (SR0140.8.9)** process
  parameter (page 156))

  ■ Value
  (taken from the [Long] **Master bundle identifier (SR0140.8.9)** process
  parameter (page 156))

- **Measured Value Bundle**:

  ■ Key
  (taken from the [Measured Value] **Master bundle identifier
  (SR0140.8.10)** process parameter (page 157))

  ■ Value
  (taken from the [Measured Value] **Master bundle identifier
  (SR0140.8.10)** process parameter (page 157))

- **String Value Bundle**:

  - Key
    (taken from the [String] **Master bundle identifier (SR0140.8.11)**
    process parameter (page 157))

  - Value
    (taken from the [String] **Master bundle identifier (SR0140.8.11)**
    process parameter (page 157))

- **Timestamp Value Bundle**:

  - Key
    (taken from the [Timestamp] **Master bundle identifier (SR0140.8.12)**
    process parameter (page 158))

  - Value
    (taken from the [Timestamp] **Master bundle identifier (SR0140.8.12)**
    process parameter (page 158))

5. **Confirm** button.

Completed mode (SR0140.1.3)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase)

2. <Instruction text>
   (taken from **Instruction (SR0140.8.1)** process parameter (page 153))

3. Context identifier (taken from **Context definition (SR0140.8.3)** process
   parameter (page 153).

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

   - **BigDecimal Value Bundle**:

     - Key
       (taken from the [BigDecimal] **Master bundle identifier (SR0140.8.6)**
       process parameter (page 155))

     - Value
       (taken from the [BigDecimal] **Master bundle identifier (SR0140.8.6)**
       process parameter (page 155))

- **Boolean Value Bundle**:

  - Key
    (taken from the [Boolean] **Master bundle identifier (SR0140.8.7)** process parameter (page 155))

  - Value
    (taken from the [Boolean] **Master bundle identifier (SR0140.8.7)** process parameter (page 155))

- **Duration Value Bundle**:

  - Key
    (taken from the [Duration] **Master bundle identifier (SR0140.8.8)** process parameter (page 156))

  - Value
    (taken from the [Duration] **Master bundle identifier (SR0140.8.8)** process parameter (page 156))

- **Long Value Bundle**:

  - Key
    (taken from the [Long] **Master bundle identifier (SR0140.8.9)** process parameter (page 156))

  - Value
    (taken from the [Long] **Master bundle identifier (SR0140.8.9)** process parameter (page 156))

- **Measured Value Bundle**:

  - Key
    (taken from the [Measured Value] **Master bundle identifier (SR0140.8.10)** process parameter (page 157))

  - Value
    (taken from the [Measured Value] **Master bundle identifier (SR0140.8.10)** process parameter (page 157))

- **String Value Bundle**:

  - Key
    (taken from the [String] **Master bundle identifier (SR0140.8.11)** process parameter (page 157))

  - Value
    (taken from the [String] **Master bundle identifier (SR0140.8.11)** process parameter (page 157))

■ **Timestamp Value Bundle**:

    ■ Key
    (taken from the [Timestamp] **Master bundle identifier (SR0140.8.12)**
    process parameter (page 158))

    ■ Value
    (taken from the [Timestamp] **Master bundle identifier (SR0140.8.12)**
    process parameter (page 158))

5. **Confirm** button (completed).

Representation in Navigator (SR0140.4+)

The Navigator provides the following details:

Phase column (Framework capability)

■ <Phase name>

    ■ Example:
    Write workflow context

Information column (SR0140.4.1)

■ <Context identifier> (taken from **Context definition (SR0140.8.3)** process parameter (page 153).

    ■ Example: WF1900002485

Action column

■ There are no phase-specific actions available.

Representation in Sub-report (SR0140.5+)

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 (SR0140.5.1)

- Instruction table panel and/or instruction link panel
  (only if an instruction table and/or instruction link is defined for the phase)

- Instruction text

- Context identifier

- Table of values that have been written (in the order of the bundle-specific process parameters):

  - **BigDecimal Value Bundles**:

    - Key

    - Value

  - **Boolean Value Bundles**:

    - Key

    - Value

  - **Duration Value Bundles**:

    - Key

    - Value

  - **Long Value Bundles**:

    - Key

    - Value

  - **Measured Value Bundles**:

    - Key

    - Value

  - **String Value Bundles**:

    - Key

    - Value

  - **Timestamp Value Bundles**:

    - Key

    - Value

Business Logic (SR0140.2+)

The phase implements the following business logic.

Phase Mode

Business logic related to phase modes.

Manual completion mode (SR0140.2.1)

- 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 (SR0140.1.2)** layout (page 140).
Phase completion	20	See **Confirm phase (SR0140.2.5)** function (page 147).

Automatic completion mode (SR0140.2.2)

- Function: **Automatic completion** mode of phase

- Type: Phase mode

- Trigger: Phase becomes active

- Postcondition: Phase is completed

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0140.1.2)** layout (page 140).
	20	The system tries to confirm the phase immediately as if the operator had used the **Confirm** button of the phase to trigger the **Confirm phase (SR0140.2.5)** function (page 147).

Main Path

Business logic related to the main path:

Confirm phase (SR0140.2.3)

- ■ Function: Completion of phase

- ■ Trigger: Operator confirms phase

- ■ Postcondition: Phase is completed

Step	#	Description
Operator confirms phase	10	Operator confirms that the context data shall be stored.
System checks context identifier	20	If no context identifier is configured with the **Context definition (SR0140.8.3)** process parameter (page 153), phase displays the **Context identifier not defined (SR0140.3.6.2)** error message (page 161). When the error message has been confirmed, phase returns to the Active mode (SR0140.1.2) layout. Otherwise continue with step 30.
System checks keys	30	If no key is configured at any of the bundle process parameters ■ [BigDecimal] Master (Bundle identifier) (SR0140.8.6) process parameter (page 155), ■ [Boolean] Master (Bundle identifier) (SR0140.8.7) process parameter (page 155), ■ [Duration] Master (Bundle identifier) (SR0140.8.8) process parameter (page 156), ■ [Long] Master (Bundle identifier) (SR0140.8.9) process parameter (page 156), ■ [MeasuredValue] Master (Bundle identifier) (SR0140.8.10) process parameter (page 157), ■ [String] Master (Bundle identifier) (SR0140.8.11) process parameter (page 157), ■ [Timestamp] Master (Bundle identifier) (SR0140.8.12) process parameter (page 158), phase displays the **Key not defined (SR0140.3.6.1)** error message (page 161). When the error message has been confirmed, phase returns to the **Active mode (SR0140.1.2)** layout (page 140). Otherwise continue with step 40.

Step	#	Description
System checks data type consistency	40	If context data for the context identifier defined by **Context definition (SR0140.8.3)** process parameter (page 153) and key defined at any of the bundle process parameters
		■ [BigDecimal] Master (Bundle identifier) (SR0140.8.6) process parameter (page 155),
		■ [Boolean] Master (Bundle identifier) (SR0140.8.7) process parameter (page 155),
		■ [Duration] Master (Bundle identifier) (SR0140.8.8) process parameter (page 156),
		■ [Long] Master (Bundle identifier) (SR0140.8.9) process parameter (page 156),
		■ [MeasuredValue] Master (Bundle identifier) (SR0140.8.10) process parameter (page 157),
		■ [String] Master (Bundle identifier) (SR0140.8.11) process parameter (page 157),
		■ [Timestamp] Master (Bundle identifier) (SR0140.8.12) process parameter (page 158),
		already exists with a different data type than that of the bundle process parameters, phase displays the **Change of data type not allowed (SR0140.3.6.3)** error message (page 161). When the error message has been confirmed, phase returns to the **Active mode (SR0140.1.2)** layout (page 140). Otherwise continue with step 50.

Step	#	Description
System checks if context data already exists	50	If check is enabled according to the **Context and key check (SR0140.8.5)** process parameter (page 153) and both context data for the context identifier defined by the **Context definition (SR0140.8.3)** process parameter (page 153) and key defined at any of the bundle process parameters ■ [BigDecimal] Master (Bundle identifier) (SR0140.8.6) process parameter (page 155), ■ [Boolean] Master (Bundle identifier) (SR0140.8.7) process parameter (page 155), ■ [Duration] Master (Bundle identifier) (SR0140.8.8) process parameter (page 156), ■ [Long] Master (Bundle identifier) (SR0140.8.9) process parameter (page 156), ■ [MeasuredValue] Master (Bundle identifier) (SR0140.8.10) process parameter (page 157), ■ [String] Master (Bundle identifier) (SR0140.8.11) process parameter (page 157), ■ [Timestamp] Master (Bundle identifier) (SR0140.8.12) process parameter (page 158), already exist, phase creates the **Context and key check (SR0140.3.2.1)** system-triggered exception (page 158). Otherwise continue with step 60.

Step	#	Description
System writes context data	60	Phase stores the context data with the context identifier defined by the **Context definition (SR0140.8.3)** process parameter (page 153) and key and value defined at any of the bundle process parameters ■ [BigDecimal] Master (Bundle identifier) (SR0140.8.6) process parameter (page 155), ■ [Boolean] Master (Bundle identifier) (SR0140.8.7) process parameter (page 155), ■ [Duration] Master (Bundle identifier) (SR0140.8.8) process parameter (page 156), ■ [Long] Master (Bundle identifier) (SR0140.8.9) process parameter (page 156), ■ [MeasuredValue] Master (Bundle identifier) (SR0140.8.10) process parameter (page 157), ■ [String] Master (Bundle identifier) (SR0140.8.11) process parameter (page 157), ■ [Timestamp] Master (Bundle identifier) (SR0140.8.12) process parameter (page 158), in the database.
	70	Phase is completed.

Process Parameters (SR0140.8+)

The following process parameters define the behavior of the phase.

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

INSTRUCTION LINK-SPECIFIC PARAMETERS

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 152). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

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

Instruction (SR0140.8.1)

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

Mode (SR0140.8.2)

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 data has been written.

Context definition (SR0140.8.3)

Attribute	Type	Comment
Identifier	String	Defines the context identifier for all key/value pairs. Maximum length is 250 characters.

Context and key check (SR0140.8.5)

Attribute	Type	Comment
Enabled	String	Controls if a check is performed.
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 **Context and key check (SR0140.3.2.1)** system-triggered exception (page 158).

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Override value (SR0140.8.4)

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 value (SR0140.3.1.1)** user-triggered exception (page 159).

BigDecimal Value Bundle

[BigDecimal] 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" [A2].

[BigDecimal] Master (Bundle identifier) (SR0140.8.6)

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	BigDecimal	Defines the value for the key and context identifier.

Boolean Value Bundle

[Boolean] 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" [A2].

[Boolean] Master (Bundle identifier) (SR0140.8.7)

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	Boolean	Defines the value for the key and context identifier.

Duration Value Bundle

[Duration] 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" [A2].

[Duration] Master (Bundle identifier) (SR0140.8.8)

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	Duration	Defines the value for the key and context identifier.

Long Value Bundle

[Long] 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" [A2].

[Long] Master (Bundle identifier) (SR0140.8.9)

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	Long	Defines the value for the key and context identifier.

Measured Value Bundle

[Measured Value] 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" [A2].

[Measured Value] Master (Bundle identifier) (SR0140.8.10)

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	MeasuredValue	Defines the value for the key and context identifier.

String Value Bundle

[String] 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" [A2].

[String] Master (Bundle identifier) (SR0140.8.11)

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	String	Defines the value for the key and context identifier.

Timestamp Value Bundle

[Timestamp] 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" [A2].

[Timestamp] Master (Bundle identifier) (SR0140.8.12)

Attribute	Type	Comment
Key	String	Defines the key that can be used with the context identifier to retrieve the value. Maximum length is 250 characters.
Value	Timestamp	Defines the value for the key and context identifier.

Exceptions (SR0140.3+)

The phase supports user-defined, user-triggered (page 117), system-triggered (page 116), and post-completion exceptions (page 117) and their configuration by means of process parameters (page 112).

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

System-triggered Exceptions (SR0140.3.2+)

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.

Context and key check (SR0140.3.2.1)

If enabled, the **Context and key check** exception checks if for the context identifier and key there is already a data set, which would consequently be updated.

Representation of the exception:

- ◼ <Exception text>
  (taken from **Context and key check (SR0140.8.5)** process parameter (page 153))
  The <Key> key already exists and will be updated.

  - ◼ Example:
    Unexpected update of existing key, review of recipe is needed.
    The washing program key already exists and will be updated.

Context and key check Logic (SR0140.3.2.1.1)

- Trigger: Operator confirms phase

- Postcondition: Exception is recorded

Step	#	Description
Operator confirms phase	10	Phase creates **Context and key check (SR0140.3.2.1)** system-triggered exception.
Operator triggers and confirms exception	20	Phase records the exception and returns to the **Active mode (SR0140.1.2) layout** (page 140).

User-triggered Exceptions (SR0140.3.1+)

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.

Override value (SR0140.3.1.1)

The **Override value** exception allows an operator to change the value for a bundle parameter that was defined in the master recipe or master workflow. Each defined bundle parameter allows a data type-specific change of the value.

Representation during exception handling:

- Instruction
  <bundle parameter key>
  Override the defined value:
  Current value <current value>
  Override value <box for a new value, in case of a boolean value bundle display options (Yes, No)>
  Confirm button.

- Exception text:
  <Exception text>
  (taken from **Override value (SR0140.8.4)** process parameter (page 154))
  Manual override of value for the <bundle parameter key> key.
  Old value: <Old value>
  New value: <New value>

  - Example:
    Review of recipe definitions required.
    Manual override of value for the washing program key.
    Old value: CleanP102
    New value: CleanP109

Override value - Logic (SR0140.3.1.1.1)

■ Trigger: Exception is selected

■ Postcondition: Changed value is available

Step	#	Description
Operator confirms exception	10	Phase checks the format of the input. In case of an input with wrong format, the phase shows an error message. Depending on the type of the bundle parameter, the following error messages are displayed: ■ **Override BigDecimal value with wrong format (SR0140.3.6.4)** error (page 161) ■ **Override Duration value with wrong format (SR0140.3.6.5)** error (page 161) ■ **Override Long value with wrong format (SR0140.3.6.6)** error (page 162) ■ **Override Measured Value with wrong format (SR0140.3.6.7)** error (page 162) ■ **Override Timestamp value with wrong format (SR0140.3.6.8)** error (page 162) If no error has occurred, phase shows the exception description to be signed according to the **Override value (SR0140.8.4)** process parameter (page 154).
Operator signs exception	20	Phase records the exception and updates the value for key and context identifier to be used.

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 (SR0140.3.6+)

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.

Key not defined (SR0140.3.6.1)

UI text	Comment
<Information that a key is not defined for a bundle parameter.>	Message pack: PhaseWriteContextData<version> Message ID: ContextDataKeyEmptyException_msg

Context identifier not defined (SR0140.3.6.2)

UI text	Comment
<Information that context identifier is not defined.>	Message pack: PhaseWriteContextData<version> Message ID: ContextIdentifierEmptyException_msg

Change of data type not allowed (SR0140.3.6.3)

UI text	Comment
<Information that an existing key exists with different data type. Change of data type is not allowed.>	Message pack: PhaseWriteContextData<version> Message ID: ContextDataExistsWrongDataTypeException_msg

Override BigDecimal value with wrong format (SR0140.3.6.4)

UI text	Comment
<Information that the override value has a wrong format.>	Message pack: PhaseWriteContextData<version> Message ID: OverrideBigDecimalValueError_ErrorMsg

Override Duration value with wrong format (SR0140.3.6.5)

UI text	Comment
<Information that the override value has a wrong format.>	Message pack: PhaseWriteContextData<version> Message ID: OverrideDurationValueError_ErrorMsg

Override Long value with wrong format (SR0140.3.6.6)

UI text	Comment
\<Information that the override value has a wrong format.>	Message pack: PhaseWriteContextDataversion> Message ID: OverrideLongValueError_ErrorMsg

Override Measured Value with wrong format (SR0140.3.6.7)

UI text	Comment
\<Information that the override value has a wrong format.>	Message pack: PhaseWriteContextData\<version> Message ID: OverrideMeasuredValueValueError_ErrorMsg

Override Timestamp Value with wrong format (SR0140.3.6.8)

UI text	Comment
\<Information that the override value has a wrong format.>	Message pack: PhaseWriteContextData\<version> Message ID: OverrideTimestampValueError_ErrorMsg

Output Variables (SR0140.9+)

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.

Context identifier (SR0140.9.1)

- Data type: String

- Usage: The output variable provides the context identifier that was used to store the keys and values and is taken from the **Context definition (SR0140.8.3)** process parameter (page 153).

Send User Notification Phase (SR0150+)

The **Send user notification** phase allows to inform an operator on the shop floor about important data that is required to complete the current process in time and with the appropriate priority or to support the operator with taking the right decisions.

Example use cases are:

■ Send a timestamp to a filling operation to display a countdown timer by which the material must be used and the filling process stopped.

■ Send a timestamp by which the cleaning process will be finished.

Each recorded value is stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 168).

Anomalies that occur during processing are covered by the phase exception handling (page 176) (e.g., missing timestamp data).

After completion, the phase displays the notification text that will be displayed in the notification panel of the target Production Execution Client and the bundle type-specific value sent with the notification.

The Navigator displays the bundle type-specific value of the notification.

Figure 13: Send user notification during execution - Timer notification

Figure 14: Send user notification during execution - Timestamp notification

Layout

The phase provides individual layouts for its representation during execution (page 166), in the Navigator (page 167), and in the sub-report (page 168).

Representation during Execution (SR0150.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0150.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0150.8.1)** process parameter (page 173)).

2. Notification text (as defined by the **Notification Configuration (SR0150.8.3)** process parameter (page 174)).

3. Notification timestamp (as defined by the **Timer notification - End time (SR0150.8.4)** process parameter (page 175) or the **Timestamp notification - Timestamp** (SR0150.8.5) process parameter (page 175).

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

Active mode (SR0150.1.2)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase).

2. <Instruction text>
   (taken from **Instruction (SR0150.8.1)** process parameter (page 173)).

3. Notification text (as defined by the **Notification Configuration (SR0150.8.3)** process parameter (page 174)).

4. Notification timestamp (as defined by the **Timer notification - End time (SR0150.8.4)** process parameter (page 175) or the **Timestamp notification - Timestamp** (SR0150.8.5) process parameter (page 175).

5. **Confirm** button.

Completed mode (SR0150.1.3)

1. Instruction table panel and/or instruction link panel
   (only if an instruction table and/or instruction link is defined for the phase).

2. <Instruction text>
   (taken from **Instruction (SR0150.8.1)** process parameter (page 173)).

3. Notification text (as defined by the **Notification Configuration (SR0150.8.3)** process parameter (page 174)).

4. Notification timestamp (as defined by the **Timer notification - End time (SR0150.8.4)** process parameter (page 175) or the **Timestamp notification - Timestamp** (SR0150.8.5) process parameter (page 175).

5. **Confirm** button (completed).

Representation in Navigator (SR0150.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- ■ <Phase name>

    - ■ Example:
      Send filling end.

Information column (SR0150.4.1)

- ■ Notification timestamp (as defined by the **Timer notification - End time (SR0150.8.4)** process parameter (page 175) or the **Timestamp notification - Timestamp** (SR0150.8.5) process parameter (page 175).

    - ■ Example: 19:26:35 5/Sep/2021 CET

Action column

- ■ There are no phase-specific actions available.

Representation in Sub-report (SR0150.5+)

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 (SR0150.5.1)

- ◼ Instruction table panel and/or instruction link panel
  (only if an instruction table and/or instruction link is defined for the phase)

- ◼ Instruction text

- ◼ Reference path

- ◼ Notification text

- ◼ Notification end time or Timestamp (depends on used notification bundle parameter)

- ◼ Alarm

- ◼ Pinned

Business Logic (SR0150.2+)

The phase implements the following business logic.

Phase Mode

Business logic related to phase modes.

Manual completion mode (SR0150.2.1)

■ 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 (SR0150.1.2)** layout (page 166).
	20	In case there is an incomplete phase configuration the phase displays the **Phase configuration incomplete (SR0150.3.6.1)** error message text (page 177) as part of the phase view.
Phase completion	30	See **Confirm phase (SR0150.2.5)** function (page 170).

Automatic completion mode (SR0150.2.2)

■ Function: **Automatic completion** mode of phase

■ Type: Phase mode

■ Trigger: Phase becomes active

■ Postcondition: Phase is completed

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0150.1.2)** layout (page 166).
	20	In case there is an incomplete phase configuration the phase displays the **Phase configuration incomplete (SR0150.3.6.1)** error message text (page 177) as part of the phase view.
	30	The system tries to confirm the phase immediately as if the operator had used the **Confirm** button of the phase to trigger the **Confirm phase (SR0150.2.5)** function (page 170).

Main Path

Business logic related to the main path:

Confirm phase (SR0150.2.3)

- ■ Function: Completion of phase

- ■ Trigger: Operator confirms phase

- ■ Postcondition: Phase is completed

Step	#	Description
Operator confirms phase	10	Operator confirms that the notification shall be sent.
Check configuration errors	12	If the phase already displays incomplete phase configuration information, it cannot be completed and displays the **Phase configuration incomplete (SR0150.3.6.1)** error message (page 177).
System builds the notification	20	The notification identifier is copied from the **Notification Bundle Configuration (SR0150.8.3)** process parameter (page 174). If it is empty, a unique value will be generated. The path of the notification is built as concatenation of the current order or workflow identifier and the reference path of the **Notification Bundle Configuration (SR0150.8.3)** process parameter (page 174). The notification text is copied from the **Notification Bundle Configuration (SR0150.8.3)** process parameter (page 174).
System extends the notification bundle type specifically	30	[Timer] Bundle: ■ As value, the timestamp defined by the **End Time (SR0150.8.4)** process parameter (page 175) is used. ■ The notification will not be displayed as an alarm. ■ The notification will be pinned. [Timestamp] Bundle: ■ As value, the timestamp defined by the **Timestamp (SR0150.8.5)** process parameter (page 175) is used. ■ The notification will be displayed as an alarm if the **Alarm (SR0150.8.6)** process parameter (page 175) is enabled. ■ The notification will be pinned if the **Pinned (SR0150.8.7)** process parameter (page 176) is enabled.

Step	#	Description
Check for notification update	35	System checks if there already exists a notification with the same notification identifier for the current order or workflow. ■ If not: system proceeds with step 40. ■ If yes: system checks if the current notification has the same notification type and reference path as the existing one. If yes, the system proceeds with step 40, which will result in a notification update. If not, the system proceeds with step 50.
	40	System sends the notification.
	50	Phase is completed.

Process Parameters (SR0150.8+)

The following process parameters define the behavior of the phase.

INSTRUCTION TABLE-SPECIFIC PARAMETERS

Instruction table definition (Framework capability)

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

Instruction table text (Framework capability)

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

INSTRUCTION LINK-SPECIFIC PARAMETERS

Instruction text with links (Framework capability)

Attribute	Type	Comment
Instruction text	HTML text	Instruction text to be displayed. For any text enclosed in curly brackets you can define a hyperlink with the **Instruction link definition** process parameter (page 173). Example: Refer to {SOP1270} for guidance. Maximum length is 2000 characters (including HTML tags).

Instruction link definition (Framework capability)

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

BASIC PARAMETERS

Instruction (SR0150.8.1)

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

Mode (SR0150.8.2)

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 data has been written.

Notification Bundle Parameters

Independent of the notification type, a **Notification Bundle Configuration (SR0150.8.3)** process parameter (page 174) is part of the definition.

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" [A2] (page 179).

Notification Configuration (SR0150.8.3)

Only one notification bundle parameter can be defined.

Attribute	Type	Comment
Notification identifier	String	Together with the order or workflow identifier, it represents a unique value and can be used to update the notification text and/or value of the same bundle type within a further notification. Maximum length is 80 characters.
Reference path	String	Defines the visibility of the notification. Allowed values are ■ Empty ■ Unit procedure ■ Unit procedure / Operation Can be defined with the **Unit Procedure and Operation Path Editor (SR3146.9.2.17.5)** (for details, see **Other Editors (SR3146.9.2.17)** in "Functional Requirement Specification Recipe and Workflow Management" [A2]). Maximum length is 1024 characters. **TIP** A notification update needs to use the same reference path and notification type as the initial notification.

Attribute	Type	Comment
Notification text	String	Defines the text to be displayed in the target Production Execution Client in the first column of the notification panel. Maximum length is 1024 characters. **TIP** HTML tags will be shown as plain text and not interpreted by the notification panel.

Timer Notification - End Time (SR0150.8.4)

Attribute	Type	Comment
Timestamp	Timestamp	Defines the timestamp by which the countdown timer must reach 0. Can be defined with a Date/Time Picker editor or an expression.

Timestamp Notification - Timestamp (SR0150.8.5)

Attribute	Type	Comment
Timestamp	Timestamp	Defines the timestamp to be displayed with the notification. Can be defined with a Date/Time Picker editor or an expression.

Timestamp Notification - Alarm (SR0150.8.6)

Attribute	Type	Comment
Enabled	Boolean	Defines whether the notification will be displayed as an alarm or not. Default setting: **No**.

Timestamp Notification - Pinned (SR0150.8.7)

Attribute	Type	Comment
Enabled	Boolean	Defines whether the notification will be pinned at the top of the notification panel list or not. Default setting: **No**.

Exceptions (SR0150.3+)

The phase supports user-defined, user-triggered (page 176), system-triggered (page 176), and post-completion exceptions (page 176) and their configuration by means of process parameters (page 112).

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

There are no user-triggered exceptions available.

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 (SR0150.3.6+)

Error messages are represented as part of the phase view or 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.

Phase configuration incomplete (SR0150.3.6.1)

UI text	Comment
A configuration error has occurred.	Message pack: PhaseSendUserNotif<version> Message ID: wrongConfiguration_ErrorMsg
There is no parameter bundle defined.	Message pack: PhaseSendUserNotif<version> Message ID: noBundles_ErrorMsg
There is no end time defined.	Message pack: PhaseSendUserNotif<version> Message ID: endTimeNotSet_ErrorMsg
There is no timestamp defined.	Message pack: PhaseSendUserNotif<version> Message ID: timestampNotSet_ErrorMsg
Please abort the phase.	Message pack: PhaseSendUserNotif<version> Message ID: startAbort_ErrorMsg
Please use the Repair function to correct the issue or abort the phase.	Message pack: PhaseSendUserNotif<version> Message ID: startRepairOrAbort_ErrorMsg

Output Variables (SR0150.9+)

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.

User notification identifier (SR0150.9.1)

- Data type: String

- Usage: The output variable provides the identifier that was sent with the notification. It can be used to update the notification.

Notification 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" [A2] (page 179).

[Timer notification] End time (SR0150.9.2)

- Data type: Timestamp

- Usage: The output variable provides the defined value of the countdown timer end time.

[Timer Notification] Timestamp (SR0150.9.3)

- Data type: Timestamp

- Usage: The output variable provides the defined timestamp value.

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-RM006C-EN-E
A2	PharmaSuite Functional Requirement Specification Recipe and Workflow Management	PSFRSRD-RM010C-EN-E

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

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
Norbert Ern	Product Owner
Jürgen Stieber	Technical Lead
Ignaz Wangler	Test Lead

Version Information

Object	Version
PharmaSuite	10.02.00
Get process value	2.1
Get text value	2.1
Show instruction text	2.0
Show document	2.0
Get choice value	1.0
Upload image	1.0
Upload PDF	1.0
Show URL	1.0
Create workflow	1.0
Write context data	1.0
Send user notification	1.0
Functional Requirement Specification	1.0

Revision History

The following tables describe the history of this document.

Changes related to the document:

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

Changes related to "Get Process Value Phase" (page 3):

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

Changes related to "Get Text Value Phase" (page 21):

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

Changes related to "Show Instruction Text Phase" (page 35):

Object	Description	Document
Representation in Navigator (SR0030.4+) - Action column (page 36)	Editorial: "There are no phase-specific actions available." No code change implemented.	1.0

Changes related to "Show Document Phase" (page 43):

Object	Description	Document
Representation in Navigator (SR0040.4+) - Action column (page 45)	Editorial: "There are no phase-specific actions available." No code change implemented.	1.0

Changes related to "Get Choice Value Phase" (page 51):

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

Changes related to "Upload Image Phase" (page 65):

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

Changes related to "Upload PDF Phase" (page 87):

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

Changes related to "Show URL Phase" (page 109):

Object	Description	Document
Representation in Navigator (SR0120.4+) - Action column (page 111)	Editorial: "There are no phase-specific actions available." No code change implemented.	1.0

Changes related to "Create Workflow Phase" (page 119):

Object	Description	Document
Representation in Navigator (SR0130.4+) - Action column (page 122)	Editorial: "There are no phase-specific actions available." No code change implemented	1.0

Changes related to "Write Context Data Phase" (page 139):

Object	Description	Document
Representation in Navigator (SR0140.4+) - Action column (page 144)	Editorial: "There are no phase-specific actions available." No code change implemented.	1.0

Changes related to "Send User Notification" (page 165):

Object	Description	Document
Send User Notification Phase (SR0150+) (page 165)	New phase	1.0

U