LISTEN.
THINK.
SOLVE.®

PharmaSuite®

EQUIPMENT AUTOMATION PHASES

RELEASE 8.4

FUNCTIONAL REQUIREMENT SPECIFICATION

PUBLICATION PSFRSEA-RM004E-EN-E-DECEMBER-2017
Supersedes publication PSFDEA-RM004D-EN-E

Allen-Bradley • Rockwell Software

Rockwell Automation

Contact Rockwell See contact information provided in your maintenance contract.

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

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

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

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

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

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

ControlNet is a registered trademark of ControlNet International.

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

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

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

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

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

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

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

Contents

Figures

Chapter 1

Introduction

This document details the requirements of the functions implemented by the phases specific to equipment automation integration. 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 (SR0341.8.1)). If a requirement's meaning is for requirement grouping only, the identifier is appended by a plus sign (e.g. Process parameters (SR0341.8+)).
For requirements with **Framework capability** as identifier, see "Functional Requirement Specification Execution Framework" for their unique identifier, [A1] (page 113).

The revision history (page 115) lists the changes made to the document with PharmaSuite 8.3 as the comparison baseline. Changes related to a requirement are marked as "Editorial", "Update", "New", or "Deleted", changes to the additional context information are marked as "Context information-related".

Typographical Conventions

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

Bold typeface — Designates user interface texts, such as

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

`Monospaced typeface` — Designates code examples.

Chapter 2

Get OPC Values Phase (SR0341+)

The G**et OPC values** phase allows to read up to 50 tag values of one equipment entity from the automation layer. It supports the following data types:

- BigDecimal Value (Double, Float, Integer),
- String Value, and
- Boolean Value: choice between Yes and No (`true` and `false`).

An example use case is:

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

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

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

The affected equipment entity, the affected properties, their values, and their timestamps are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 7).

Anomalies that occur during processing are covered by the phase exception handling (page 19) (e.g. equipment entity is not available).

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

The Navigator displays the identifier of the affected equipment entity.

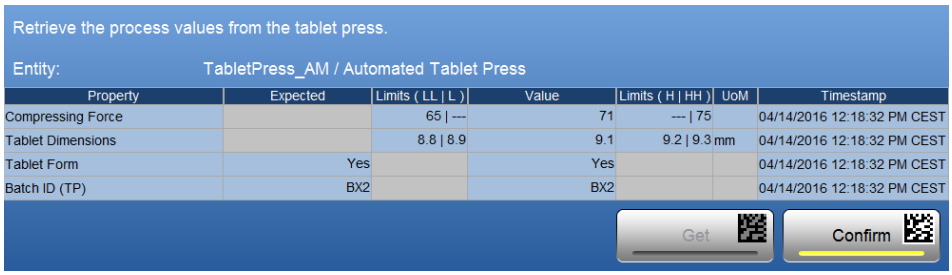

Figure 1: Get OPC values during execution

Layout

The phase provides individual layouts for its representation during execution (page 4), in the Navigator (page 6), and in the sub-report (page 7).

Representation during Execution (SR0341.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0341.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0341.8.1)** process parameter (page 12))
2. Entity:
3. List of up to 50 property types in the order of the property-specific process parameters:
   - **Numeric Property Bundle:**
     List of numeric properties
     (taken from **Numeric property - Master (bundle identifier) (SR0341.8.5)** process parameter (page 13))
   - **String Property Bundle:**
     List of string properties
     (taken from **String property - Master (bundle identifier) (SR0341.8.9)** process parameter (page 16))
   - **Boolean Property Bundle:**
     List of boolean properties
     (taken from **Boolean property - Master (bundle identifier) (SR0341.8.12)** process parameter (page 18))
4. **Get** button (disabled).
5. **Confirm** button (disabled).

Active mode (SR0341.1.2)

1. <Instruction text>
   (taken from **Instruction (SR0341.8.1)** process parameter (page 12))
2. Entity: <equipment entity identifier> / <equipment entity short description>
   (taken from **Identified equipment entity (SR0341.8.2)** process parameter (page 12))
3. List of up to 50 property types in the order of the property-specific process parameters:
   - **Numeric Property Bundle**:
     List of numeric properties
     (taken from **Numeric property - Master (bundle identifier) (SR0341.8.5)** process parameter (page 13))
     - For the representation of the value, see **Get values (SR0341.2.3)** function (page 9).
     - Last change timestamp per tag from automation layer or **Manual** in case a value has been overridden by using the **Override recorded value (SR0341.3.1.1)** user-triggered exception (page 21).
   - **String Property Bundle**:
     List of string properties
     (taken from **String property - Master (bundle identifier) (SR0341.8.9)** process parameter (page 16))
     - For the representation of the value, see **Get values (SR0341.2.3)** function (page 9).
     - Last change timestamp per tag from automation layer or **Manual** in case a value has been overridden by using the **Override recorded value (SR0341.3.1.2)** user-triggered exception (page 23).
   - **Boolean Property Bundle**:
     List of boolean properties
     (taken from **Boolean property - Master (bundle identifier) (SR0341.8.12)** process parameter (page 18))
     - For the representation of the value, see **Get values (SR0341.2.3)** function (page 9).
     - Last change timestamp per tag from automation layer or **Manual** in case a value has been overridden by using the **Override recorded value (SR0341.3.1.3)** user-triggered exception (page 24).
4. **Get** button.
5. **Confirm** button.

Completed mode (SR0341.1.3)

1. <Instruction text>
   (taken from **Instruction (SR0341.8.1)** process parameter (page 12))
2. Entity: <equipment entity identifier> / <equipment entity short description>
   (taken from **Identified equipment entity (SR0341.8.2)** process parameter (page 12))
3. List of up to 50 property types in the order of the property-specific process parameters:
   - **Numeric Property Bundle**:
     List of numeric properties
     (taken from **Numeric property - Master (bundle identifier) (SR0341.8.5)** process parameter (page 13))
     - Last change timestamp per tag from automation layer or **Manual** in case a value has been overridden by using the **Override recorded value (SR0341.3.1.1)** user-triggered exception (page 21).
   - **String Property Bundle**:
     List of string properties
     (taken from **String property - Master (bundle identifier) (SR0341.8.9)** process parameter (page 16))
     - Last change timestamp per tag from automation layer or **Manual** in case a value has been overridden by using the **Override recorded value (SR0341.3.1.2)** user-triggered exception (page 23).
   - **Boolean Property Bundle**:
     List of boolean properties
     (taken from **Boolean property - Master (bundle identifier) (SR0341.8.12)** process parameter (page 18))
     - Last change timestamp per tag from automation layer or **Manual** in case a value has been overridden by using the **Override recorded value (SR0341.3.1.3)** user-triggered exception (page 24).
4. **Get** button (disabled).
5. **Confirm** button (completed).

Representation in Navigator (SR0341.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>
  - Example:
    Get mixer values

Information column (SR0341.4.1)

- <Identifier of affected equipment entity>
  - Example: MixerA12

Action column

- There are no actions available.

Representation in Sub-report (SR0341.5+)

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 (SR0341.5.1)

- Instruction text
- Entity (identifier and short description)
- Table of values that have been read during execution (in the order of the property-specific process parameters).
  - List of numeric properties
    - Identifier
    - Limits (LL | L)
    - Value
    - Limits (H | HH)
    - UoM
    - Timestamp

- List of string properties
  - Identifier
  - Expected value
  - Value
  - Timestamp
- List of boolean properties
  - Identifier
  - Expected value
  - Timestamp

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

Business Logic (SR0341.2+)

The phase implements the following business logic.

Phase Mode

Business logic related to phase modes.

Manual completion mode (SR0341.2.1)

- 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 (SR0341.1.2)** layout (page 5).
Operator interaction	20	The **Get** button reads the tag values, see **Get values (SR0341.2.3)** function (page 9). Each time the **Get** button is used, all of the tag values are read unless they have already been read or overridden.
Phase completion	30	See **Confirm phase (SR0341.2.4)** function (page 11).

Automatic completion mode (SR0341.2.2)

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

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0341.1.2)** layout (page 5).
Phase gets values	20	See **Get values (SR0341.2.3)** function (page 9). ■ If no error has occurred, continue with the **Confirm phase (SR0341.2.4)** function (page 11). ■ If an error or warning has occurred, phase must be completed manually. See **Manual completion (SR0341.2.1)** mode (page 8).

Main Path

Business logic related to the main path:

Get values (SR0341.2.3)

- Function: Read tag values
- Type: Main path
- Trigger: Operator gets values or **Automatic completion (SR0341.2.2)** mode (page 9) is active
- Postcondition: Phase is active

Step	#	Description
Phase checks manual override	10	If a value has been overridden with the **Override recorded value (Numeric property) (SR0341.3.1.1)** user-triggered exception (page 21), **Override recorded value (String property) (SR0341.3.1.2)** user-triggered exception (page 23), or **Override recorded value (Boolean property) (SR0341.3.1.3)** user-triggered exception (page 24) and the exception has been signed, the **Get** action cannot be executed for such a value; phase displays **Override value recorded (SR0341.3.4.1)** information message (page 27).
Phase checks for read tag values	15	If a tag value has already been read from the automation layer, the **Get** action is not executed for the value.

Step	#	Description
Phase gets values	20	Phase reads the remaining tag values and disables the **Get** button as soon as there are no tag values that have not yet been read or overridden. The order of the **Numeric property - Master (bundle identifier) (SR0341.8.5)** process parameters (page 13), **String property - Master (bundle identifier) (SR0341.8.9)** process parameters (page 16), and **Boolean property - Master (bundle identifier) (SR0341.8.12)** process parameters (page 18) defines the read sequence of property tag values. Process parameters without property types are skipped. If one of the following issues occurs, phase behavior is as follows:
Property cannot be read due to a pre-reading issue	20.1	■ Phase does not display a value, ■ changes cell background to red, ■ appends "(X)" to the "empty value", and ■ displays **Invalid configuration error (SR0341.3.6.1)** error message (page 28).
Property cannot be read due to an automation integration issue or tag data quality is rated as bad	20.2	■ Phase does not display a value, ■ changes cell background to red, ■ appends "(X)" to the "empty value", and ■ displays **System error (SR0341.3.6.4)** error message (page 28), **No get result error (SR0341.3.6.5)** error message (page 29), or **Automation error (SR0341.3.6.3)** error message (page 28).
Validation	20.3	➢ **Numeric Property Bundle** Phase checks the numeric value against the settings of the **Limit definition (SR0341.8.8)** process parameter (page 15). Limits are checked in the following order: LL/HH » L/H. If the check is violated, phase creates the **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19) for a numeric value. ➢ **String Property Bundle** Phase checks the string value against the settings of the **Expected value definition (SR0341.8.11)** process parameter (page 17). If the check is violated, phase creates the **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19) for a string value. ➢ **Boolean Property Bundle** Phase checks the boolean value against the settings of the **Expected value definition (SR0341.8.14)** process parameter (page 18). If the check is violated, phase creates the **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19) for a boolean value.

Step	#	Description
		If a check is violated, phase changes cell background to yellow. After the exception has been signed, phase changes cell background to the default and adds the exception marker to the value's cell. If no check is violated, phase returns to the **Active mode (SR0341.1.2)** layout (page 5).
	30	If applicable, continue with the **Override recorded value (Numeric property) (SR0341.3.1.1)** user-triggered exception (page 21), **Override recorded value (String property) (SR0341.3.1.2)** user-triggered exception (page 23), or **Override recorded value (Boolean property) (SR0341.3.1.3)** user-triggered exception (page 24). Phase can be completed with the **Confirm phase (SR0341.2.4)** function (page 11).

Confirm phase (SR0341.2.4)

- Function: Completion of phase
- Type: Main path
- Trigger: Operator confirms phase or **Automatic completion (SR0341.2.2)** mode (page 9) is active
- Postcondition: Phase is completed

Step	#	Description
In **Manual completion (SR0341.2.1)** mode (page 8): Operator confirms phase	10	Operator confirms the tag values.
Phase performs completion checks	20	If one of the following issues occurs, phase cannot be completed: ■ In **Manual completion (SR0341.2.1)** mode (page 8), the **Get** button has not been used. ■ Not all values whose tags are enabled have been read. Phase displays **Recorded values incomplete (SR0341.3.6.7)** error message (page 29). If a validation check fails, phase creates the **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19).
Phase completion	30	Phase is completed.

Process Parameters (SR0341.8+)

The following process parameters define the behavior of the phase.

BASIC PARAMETERS

Instruction (SR0341.8.1)

➢ For recent changes, see revision history (page 115).

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

Identified equipment entity (SR0341.8.2)

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

Mode (SR0341.8.3)

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

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Override recorded value (SR0341.8.4)

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)**.

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

See also **Override recorded value (Numeric property bundle) (SR0341.3.1.1)** user-triggered exception (page 21), **Override recorded value (Boolean property bundle) SR0341.3.1.3)** user-triggered exception (page 24), and **Override recorded value (String property bundle) (SR0341.3.1.2)** user-triggered exception (page 23).

Numeric Property Bundle

Bundle process parameters (Framework capability)

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

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

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A2] (page 113).

Master (Bundle identifier) (SR0341.8.5)

Attribute	Type	Comment
Property	String	Equipment property to be read.

Property Selection editor (Framework capability)

The system provides a Property Selection editor for selecting an equipment property based on its data type (numeric, string, boolean).

L-H configuration (SR0341.8.6)

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

1. LL-HH (defined with the **LL-HH configuration (SR0341.8.7)** process parameter (page 14))
2. L-H

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **L limit** or **H limit** attributes of the **Limit definition** process parameter (page 15) are set. If they are not set, the validation will fail. Default setting: **No**.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

See also **Override recorded value (SR0341.3.1.1)** user-triggered exception (page 21) and **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19).

LL-HH configuration (SR0341.8.7)

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

1. LL-HH
2. L-H (defined with the **L-H configuration (SR0341.8.6)** process parameter (page 14))

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **LL limit** or **HH limit** attributes of the **Limit definition** process parameter (page 15) are set. If they are not set, the validation will fail. Default setting: **No**.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

See also **Override recorded value (SR0341.3.1.1)** user-triggered exception (page 21) and **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19).

Limit definition (SR0341.8.8)

The following rule applies to the attributes:

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

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

Attribute	Type	Comment
HH limit	BigDecimal (Double, Float, Integer)	are truncated at the end in the Phase Preview of Recipe and Workflow Designer and Production Execution Client.

See also **Override recorded value (SR0341.3.1.1)** user-triggered exception (page 21) and **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19).

String Property Bundle

Bundle process parameters (Framework capability)

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

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

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A2] (page 113).

Master (Bundle identifier) (SR0341.8.9)

Attribute	Type	Comment
Property	String	Equipment property to be read.

Property Selection editor (Framework capability)

The system provides a Property Selection editor for selecting an equipment property based on its data type (numeric, string, boolean).

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Expected value configuration (SR0341.8.10)

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **Value** attribute of the **Expected value definition** process parameter (page 17) is set. If it is not set, the validation will fail. Default setting: **No**.

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

See also **Override recorded value (SR0341.3.1.2)** user-triggered exception (page 23) and **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19).

Expected value definition (SR0341.8.11)

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

See also **Override recorded value (SR0341.3.1.2)** user-triggered exception (page 23) and **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19).

Boolean Property Bundle

Bundle process parameters (Framework capability)

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

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

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A2] (page 113).

Master (Bundle identifier) (SR0341.8.12)

Attribute	Type	Comment
Property	String	Equipment property to be read.

Property Selection editor (Framework capability)

The system provides a Property Selection editor for selecting an equipment property based on its data type (numeric, string, boolean).

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Expected value configuration (SR0341.8.13)

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, ensure that the **Value** attribute of the **Expected value definition** process parameter (page 18) is set. If it is not set, the validation will fail. Default setting: **No**.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 250 characters.

See also **Override recorded value (SR0341.3.1.3)** user-triggered exception (page 24) and **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19).

Expected value definition (SR0341.8.14)

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

See also **Override recorded value (SR0341.3.1.3)** user-triggered exception (page 24) and **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19).

Exceptions (SR0341.3+)

The phase supports user-defined, user-triggered (page 21), system-triggered (page 19), and post-completion exceptions (page 26) and their configuration by means of process parameters (page 12).

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

System-triggered Exceptions (SR0341.3.2+)

➢ For recent changes, see revision history (page 115).

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 (SR0341.3.2.1)

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

Representation of the exception:

- A violation of limits or expected values has occurred.
- List of up to 50 property types in the order of the property-specific process parameters:
  - **Numeric Property Bundle**:
    Exception text:
    <Exception text>
    (taken from **L-H configuration (SR0341.8.6)** process parameter (page 14) or **LL-HH configuration (SR0341.8.7)** process parameter (page 14))
    Property: <property identifier>
    <Affected limit, L, LL, H, HH>: <limit value>
    (taken from **Limit definition (SR0341.8.8)** process parameter (page 15))
    Actual value: <OPC value>
    - Example:
      Limit violation confirmed.
      Property: AgitatorSpeed

LL limit: 300 rpm
Actual value: 200 rpm

- **String Property Bundle**:
  Exception text:
  <Exception text>
  (taken from **Expected value configuration (SR0341.8.10)** process parameter (page 16))
  Property: <property identifier>
  Expected value: <expected value>
  (taken from **Expected value definition (SR0341.8.11)** process parameter (page 17))
  Actual value: <OPC value>
  - Example:
    Expected value violation confirmed.
    Property: VisualCheckResult
    Expected value: Dark blue
    Actual value: Light blue
- **Boolean Property Bundle**:
  Exception text:
  <Exception text>
  (taken from **Expected value configuration (SR0341.8.13)** process parameter (page 18))
  Property: <property identifier>
  Expected value <expected value>
  (taken from **Expected value definition (SR0341.8.14)** process parameter (page 18))
  Actual value: <OPC value>
  - Example:
    Expected value violation confirmed.
    Property: HeatingPerformed
    Expected value: Yes
    Actual value: No

Limit violation - Logic (SR0341.3.2.1.1)

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

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

Step	#	Description
Operator signs exception	20	Phase records exception.

Multiple system-triggered exceptions (SR0341.3.2.2)

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

User-triggered Exceptions (SR0341.3.1+)

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.

Numeric Property Bundle

Override recorded value (SR0341.3.1.1)

The **Override recorded value** exception allows an operator to override the numeric value read from the entity.

There is one exception per numeric property.
Properties for which an error has been detected are displayed at the top of the list properties.

Representation during exception handling:

- Instruction:
  Override recorded values:
  Current value: <current value> <UoM>
  Override value: <new value> <UoM>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override recorded value (SR0341.8.4)** process parameter (page 12))
  Property: <property identifier>
  Old value: <value> <UoM>
  New value: <value> <UoM>
  - Example:
    Value overridden.
    Property: AgitatorSpeed

Old value: 12.43 rpm
New value: 12.93 rpm

Override recorded value - Logic (SR0341.3.1.1.1)

➢ For recent changes, see revision history (page 115).

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

Step	#	Description
Operator triggers exception	10	Phase displays Exception Window.
	20	Operator enters values. If the following issue occurs, phase displays an error message: ■ Data format does not match, **Invalid data format error (SR0341.3.6.8)** error message (page 30).
Operator confirms exception	30	If the related check is enabled, phase checks the numeric value against the settings of the **Limit definition (SR0341.8.8)** process parameter (page 15). Limits are checked in the following order: LL/HH » L/H.
	30.1	If a limit is violated, phase displays a corresponding message dialog with an **Exception** button, the exception text (taken from **L-H configuration (SR0341.8.6)** process parameter (page 14) or **LL-HH configuration (SR0341.8.7)** process parameter (page 14)), the property identifier, the affected limit, and the actual value.
Operator accepts exceptional situation	30.1.1	Phase displays only one combined exception (user-triggered exception), including both exception texts from the override-value-exception and from the limit violation (see **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19) for a numeric value).
Operator cancels exceptional situation	30.1.2	Phase requires the operator to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).
	30.2	If no limit is violated or no check applies, the override value-related exception is displayed.
	30.3	If the following issue occurs, phase displays an error message: ■ Override value is missing, **No value overridden (SR0341.3.6.9)** error message (page 30). Phase shows exception description to be signed according to **Override recorded value (SR0341.8.4)** process parameter (page 12).

Step	#	Description
Operator signs exception	40	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0341.1.2)** layout (page 5).

STRING PROPERTY BUNDLE

Override recorded value (SR0341.3.1.2)

The **Override recorded value** exception allows an operator to override the string value read from the entity.

There is one exception per string property.
Properties for which an error has been detected are displayed at the top of the list properties.

Representation during exception handling:

- Instruction:
  Override recorded value
  Current value: <current value>
  Override value: <value>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override recorded value (SR0341.8.4)** process parameter (page 12))
  Property: <property identifier>
  Old value: <old value>
  New value: <new value>
  - Example:
    Value overridden.
    Property: HeatingPerformed
    Old value: Temperature alarm
    New value: High temperature alarm

Override recorded value - Logic (SR0341.3.1.2.1)

➢ For recent changes, see revision history (page 115).

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

Step	#	Description
Operator triggers exception	10	Phase displays Exception Window.
	20	Operator enters value.
Operator confirms exception	30	If the related check is enabled, phase checks the string value against the settings of the **Expected value definition (SR0341.8.11)** process parameter (page 17).
	30.1	If the expected value is violated, phase displays a corresponding message dialog with an **Exception** button, the exception text (taken from **Expected value configuration (SR0341.8.10)** process parameter (page 16)), the property identifier, the affected limit, and the actual value.
Operator accepts exceptional situation	30.1.1	Phase displays only one combined exception (user-triggered exception), including both exception texts from the override-value-exception and from the violation of the expected value (see **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19) for a string value).
Operator cancels exceptional situation	30.1.2	Phase requires the operator to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).
	30.2	If the expected value is not violated or no check applies, the override value-related exception is displayed.
	30.3	If the following issue occurs, phase displays an error message: ■ Override value is missing, **No value overridden (SR0341.3.6.10)** error message (page 30). Phase shows exception description to be signed according to **Override recorded value (SR0341.8.4)** process parameter (page 12).
Operator signs exception	40	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0341.1.2)** layout (page 5).

BOOLEAN PROPERTY BUNDLE

Override recorded value (SR0341.3.1.3)

The **Override recorded value** exception allows an operator to override the boolean value read from the entity.

There is one exception per boolean property.
Properties for which an error has been detected are displayed at the top of the list properties.

Representation during exception handling:

- Instruction:
  Override recorded value
  Current value: <current value>
  Override value: <value>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override recorded value (SR0341.8.4)** process parameter (page 12))
  Property: <property identifier>
  Old value: <old value>
  New value: <new value>
  - Example:
    Value overridden.
    Property: Infrared sensor on
    Old value: True
    New value: False

Override recorded value - Logic (SR0341.3.1.3.1)

➢ For recent changes, see revision history (page 115).

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

Step	#	Description
Operator triggers exception	10	Phase displays Exception Window.
	20	Operator selects value.
Operator confirms exception	30	If the check is enabled, phase checks the boolean value against the settings of the **Expected value definition (SR0341.8.14)** process parameter (page 18).
	30.1	If the expected value is violated, phase displays a corresponding message dialog with an **Exception** button, the exception text (taken from **Expected value configuration (SR0341.8.13)** process parameter (page 18)), the property identifier, the affected limit, and the actual value.
Operator accepts exceptional situation	30.1.1	Phase displays only one combined exception (user-triggered exception), including both exception texts from the override-value-exception and from the violation of the expected value (see **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19) for a boolean value).

Step	#	Description
Operator cancels exceptional situation	30.1.2	Phase requires the operator to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).
	30.2	If the expected value is not violated or no check applies, the override value-related exception is displayed.
	30.3	If the following issue occurs, phase displays an error message: ■ Override value is missing, **No value overridden (SR0342.3.6.11)** error message (page 31). Phase shows exception description to be signed according to **Override recorded value (SR0341.8.4)** process parameter (page 12).
Operator signs exception	40	Phase records exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0341.1.2)** layout (page 5).

NOT BUNDLE-SPECIFIC

Multiple exceptions (SR0341.3.1.4)

In case an **Override recorded value (Numeric property) (SR0341.3.1.1)** user-triggered exception (page 21), **Override recorded value (String property) (SR0341.3.1.2)** user-triggered exception (page 23), or **Override recorded value (Boolean property) (SR0341.3.1.3)** user-triggered exception (page 24) coincides with the **Limit violation (SR0341.3.2.1)** system-triggered exception (page 19), only one combined exception (user-triggered exception) is recorded including information about all related exceptions. The highest risk assessment of all related exceptions and its related signature privilege apply.

Post-completion Exceptions

There are no post-completion exceptions available.

Information Messages (SR0341.3.4+)

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

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

Override value recorded (SR0341.3.4.1)

UI text	Comment
1. The Get action was not successful. 2. <empty string> 3. Values have already been overridden manually. 4. <list of tags>	1. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: ExecutionError_HeaderMsg 2. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: BusinessLogic_ErrorCategory 3. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: ValuesOverridden_WarningCategory 4. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: ReadPostcheck_NoGetAfterOverride_ErrorMsg Potential error cause: **Get** button is used after the **Override recorded value (Numeric property) (SR0341.3.1.1)** user-triggered exception (page 21), **Override recorded value (String property) (SR0341.3.1.2)** user-triggered exception (page 23), or **Override recorded value (Boolean property) (SR0341.3.1.3)** user-triggered exception (page 24) has been signed.

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

Questions

There are no questions available.

Decisions

There are no decisions available.

Error Messages (SR0341.3.6+)

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

They are composed of up to three levels:

1. header,
2. category, and
3. details (not always used).

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

Get Property-specific Error Messages (Pre-reading)

Invalid configuration error (SR0341.3.6.1)

UI text	Comment
1. The Get action was not successful. 2. Please record the values manually.	1. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: CheckBeforeExecuteError_HeaderMsg 2. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: IrreparableExecution_ErrorCategory Potential error cause: ■ Tag is enabled, but the tag path is undefined. ■ The property to be read is not defined for the identified equipment entity.

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

Automation error (SR0341.3.6.3)

UI text	Comment
1. The Get action was not successful. 2. Please record the values manually.	1. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: ExecutionError_HeaderMsg 2. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: IrreparableExecution_ErrorCategory Potential error cause: ■ Referenced equipment entity is undefined (Null). ■ The quality of the read tag value is rated as bad.

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

Get Property-specific Error Messages (Reading)

System error (SR0341.3.6.4)

UI text	Comment
1. The Get action was not successful. 2. Please record the values manually.	1. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: ExecutionError_HeaderMsg 2. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: Other_ErrorCategory Potential error cause: ■ Automation Integration server cannot be reached.

UI text	Comment
	■ Automation Integration server read failure. ■ Live Data server read failure. ■ The quality of the read tag value is rated as bad.

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

No get result error (SR0341.3.6.5)

UI text	Comment
1. The Get action was not successful. 2. Please record the values manually.	1. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: ExecutionError_HeaderMsg 2. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: IrreparableExecution_ErrorCategory Potential error cause: ■ The return value of the Automation Integration server does not contain an entry for at least one defined and valid property tag path. ■ The quality of the read tag value is rated as bad. ■ Health, simulation, or maintenance verification failed.

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

Phase Completion-specific Error Messages

Recorded values incomplete (SR0341.3.6.7)

UI text	Comment
1. Cannot confirm. 2. Not all expected values have been recorded. Please record the values manually.	1. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: CompletionError_HeaderMsg 2. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: ReparableConfirm_ErrorCategory Potential error cause: The values of the enabled tags were not read successfully or overridden. Expected values are still missing.

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

User-triggered Exception-specific Error Messages

Invalid data format error (SR0341.3.6.8)

- Applies to **Numeric Property Bundle** only

UI text	Comment
1. Cannot confirm the overridden values. 2. <empty string>	1. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: OverrideExceptionConfirmationError_HeaderMsg 2. Message pack: PhaseEqmAIGetOPCValues<version> Message ID: Other_ErrorCategory Potential error cause: The entered text value cannot be converted to a numeric value of the targeted numeric data type.

NUMERIC PROPERTY BUNDLE

No value overridden (SR0341.3.6.9)

UI text	Comment
1. Enter an override value.	1. Message pack: PhaseEqmAI<version> Message ID: OverrideNumericValueNotSet_ErrorMsg Potential error cause: No override value was entered before the user-triggered exception was confirmed.

STRING PROPERTY BUNDLE

No value overridden (SR0341.3.6.10)

UI text	Comment
1. You have to enter an override value before you can confirm.	1. Message pack: PhaseEqmAI<version> Message ID: OverrideStringValueNotSet_ErrorMsg Potential error cause: No override value was entered before the user-triggered exception was confirmed.

BOOLEAN PROPERTY BUNDLE

No value overridden (SR0341.3.6.11)

UI text	Comment
1. You have to select a value before you can confirm.	1. Message pack: PhaseEqmAI<version> Message ID: OverrideBooleanValueNotSet_ErrorMsg Potential error cause: No override value was selected before the user-triggered exception was confirmed.

Output Variables (SR0341.9+)

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.

Automation get successful (SR0341.9.1)

- Data type: Boolean
- Values: `true`, `false`
- Usage: The output variable states if the get operation from the automation layer was successful.
  - The value is `true` if all property values have been read successfully.
  - The value is `false` if at least one of the property values could not be read from the automation layer or has been overridden by using the **Override recorded value (Numeric property) (SR0341.3.1.1)** user-triggered exception (page 21), **Override recorded value (String property) (SR0341.3.1.2)** user-triggered exception (page 23), or **Override recorded value (Boolean property) (SR0341.3.1.3)** user-triggered exception (page 24).

Numeric Property 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 113).

Value (SR0341.9.2)

- Data type: BigDecimal
- Usage: The output variable provides the actual value of the numeric property tag as a **BigDecimal** value. The value is Null if N/A is the phase result.

Unit of measure (SR0341.9.3)

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

Automation get successful (SR0341.9.4)

- Data type: Boolean
- Values: `true`, `false`

- Usage: The output variable states if the get operation from the automation layer was successful.
  - The value is `true` if the property value of the numeric property has been read successfully.
  - The value is `false` if the property value of the numeric property could not be read from the automation layer or has been overridden by using the **Override recorded value (SR0341.3.1.1)** user-triggered exception (page 21).

String Property 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 113).

Value (SR0341.9.6)

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

Automation get successful (SR0341.9.7)

- Data type: Boolean
- Values: `true`, `false`
- Usage: The output variable states if the get operation from the automation layer was successful.
  - The value is `true` if the property value of the string property has been read successfully.
  - The value is `false` if the property value of the string property could not be read from the automation layer or has been overridden by using the **Override recorded value (SR0341.3.1.2)** user-triggered exception (page 23).

Boolean Property 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 113).

Value (SR0341.9.9)

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

Automation get successful (SR0341.9.10)

- Data type: Boolean
- Values: `true`, `false`
- Usage: The output variable states if the get operation from the automation layer was successful.
  - The value is `true` if the property value of the boolean property has been read successfully.
  - The value is `false` if the property value of the boolean property could not be read from the automation layer or has been overridden by using the **Override recorded value (SR0341.3.1.3)** user-triggered exception (page 24).

Performance (SR0341.12+)

Performance of Get Activity (SR0341.12.1)

The time for getting the OPC values on the automation layer does not take longer than 5 seconds. Any potential delay by the OPC server or the PLC communication is not considered.

Set OPC Values Phase (SR0342+)

The **Set OPC values** phase allows to write up to 50 tag values of one equipment entity to the automation layer. It supports the following data types:

- BigDecimal Value (Double, Float, Integer) with low and high limits,
- String Value, and
- Boolean Value: choice between Yes and No (`true` and `false`).

An example use cases is:

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

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

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

The affected equipment entity, the affected properties, and their values are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 39).

Anomalies that occur during processing are covered by the phase exception handling (page 47) (e.g. equipment entity is not available).

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

The Navigator displays the identifier of the affected equipment entity.

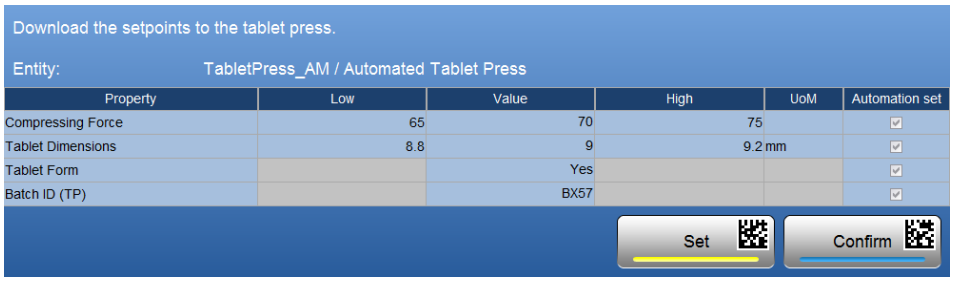

Figure 2: Set OPC values during execution

Layout

The phase provides individual layouts for its representation during execution (page 36), in the Navigator (page 38), and in the sub-report (page 39).

Representation during Execution (SR0342.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0342.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0342.8.1)** process parameter (page 43))
2. Entity:
3. List of up to 50 property types in the order of the property-specific process parameters:
   - **Numeric Property Bundle:**
     List of numeric properties
     (taken from **Numeric property - Master (bundle identifier) (SR0342.8.6)** process parameter (page 45))
     - Read-only checkbox to indicate if the values have been successfully set on the automation layer.
   - **String Property Bundle:**
     List of string properties
     (taken from **String property - Master (bundle identifier) (SR0342.8.7)** process parameter (page 46))
     - Read-only checkbox to indicate if the values have been successfully set on the automation layer.
   - **Boolean Property Bundle:**
     List of boolean properties
     (taken from **Boolean property - Master (bundle identifier) (SR0342.8.8)** process parameter (page 47))

- Read-only checkbox to indicate if the values have been successfully set on the automation layer.

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

Active mode (SR0342.1.2)

1. <Instruction text>
   (taken from **Instruction (SR0342.8.1)** process parameter (page 43))
2. Entity: <equipment entity identifier> / <equipment entity short description>
   (taken from **Identified equipment entity (SR0342.8.2)** process parameter (page 44))
3. List of up to 50 property types in the order of the property-specific process parameters:
   - **Numeric Property Bundle:**
     List of numeric properties
     (taken from **Numeric property - Master (bundle identifier) (SR0342.8.6)** process parameter (page 45))
     - For the representation of the value, see **Set values (SR0342.2.3)** function (page 41).
     - Read-only checkbox to indicate if the values have been successfully set on the automation layer.
   - **String Property Bundle:**
     List of string properties
     (taken from **String property - Master (bundle identifier) (SR0342.8.7)** process parameter (page 46))
     - For the representation of the value, see **Set values (SR0342.2.3)** function (page 41).
     - Read-only checkbox to indicate if the values have been successfully set on the automation layer.
   - **Boolean Property Bundle:**
     List of boolean properties
     (taken from **Boolean property - Master (bundle identifier) (SR0342.8.8)** process parameter (page 47))
     - For the representation of the value, see **Set values (SR0342.2.3)** function (page 41).
     - Read-only checkbox to indicate if the values have been successfully set on the automation layer.

4. **Set** button.
5. **Confirm** button.

Completed mode (SR0342.1.3)

1. <Instruction text>
   (taken from **Instruction (SR0342.8.1)** process parameter (page 43))
2. Entity: <equipment entity identifier> / <equipment entity short description>
   (taken from **Identified equipment entity (SR0342.8.2)** process parameter (page 44))
3. List of up to 50 property types in the order of the property-specific process parameters:
   - **Numeric Property Bundle:**
     List of numeric properties
     (taken from **Numeric property - Master (bundle identifier) (SR0342.8.6)** process parameter (page 45))
     - Read-only checkbox to indicate if the values have been successfully set on the automation layer.
   - **String Property Bundle:**
     List of string properties
     (taken from **String property - Master (bundle identifier) (SR0342.8.7)** process parameter (page 46))
     - Read-only checkbox to indicate if the values have been successfully set on the automation layer.
   - **Boolean Property Bundle:**
     List of boolean properties
     (taken from **Boolean property - Master (bundle identifier) (SR0342.8.8)** process parameter (page 47))
     - Read-only checkbox to indicate if the values have been successfully set on the automation layer.
4. **Set** button (disabled).
5. **Confirm** button (completed).

Representation in Navigator (SR0342.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>
  - Example:
    Set up mixer

Information column (SR0342.4.1)

- <Identifier of affected equipment entity>
  - Example: MixerA12

Action column

- There are no actions available.

Representation in Sub-report (SR0342.5+)

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 (SR0342.5.1)

- Instruction text
- Entity (identifier and short description)
- Table of values that have been set during execution (in the order of the property-specific process parameters).
  - List of numeric properties
    - Identifier
    - Low
    - Value
    - High
    - UoM
    - Value successfully set on the automation layer (yes, no)

- List of string properties
  - Identifier
  - Value
  - Value successfully set on the automation layer (yes, no)
- List of boolean properties
  - Identifier
  - Value
  - Value successfully set on the automation layer (yes, no)

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

Business Logic (SR0342.2+)

The phase implements the following business logic.

Phase Mode

Business logic related to phase modes.

Manual completion mode (SR0342.2.1)

- 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 (SR0342.1.2)** layout (page 37). If any issue related to automation is detected during phase activation, ■ phase changes the cell background to red and ■ appends "(X)" to the "empty value".
Operator interaction	20	The **Set** button writes the tag values, see **Set values (SR0342.2.3)** function (page 41). Each time the **Set** button is used, all of the tag values are written.
Phase completion	30	See **Confirm phase (SR0342.2.4)** function (page 43).

Automatic completion mode (SR0342.2.2)

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

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0342.1.2)** layout (page 37). If any issue related to automation is detected during phase activation, ■ phase changes the cell background to red and ■ appends "(X)" to the "empty value".
Phase sets values	20	See **Set values (SR0342.2.3)** function (page 41). ■ If no error has occurred, continue with the **Confirm phase (SR0342.2.4)** function (page 43). ■ If at least one of the values could not be set automatically, phase must be completed manually. See **Manual completion (SR0342.2.1)** mode (page 40).

Main Path

Business logic related to the main path:

Set values (SR0342.2.3)

- Function: Write tag values
- Type: Main path
- Trigger: Operator sets values or **Automatic completion (SR0342.2.2)** mode (page 41) is active
- Postcondition: Phase is active

Step	#	Description
Phase checks for "input at equipment"	10	If the **Input at equipment (SR0342.3.1.1)** user-triggered exception (page 48) has been signed before, the **Set** action cannot be executed; phase displays **Input at equipment recorded (SR0342.3.6.2)** error message (page 53).

Step	#	Description
Phase sets values	20	Phase writes the tag values. The order of the **Numeric property - Master (bundle identifier) (SR0342.8.6)** process parameters (page 45), **String property - Master (bundle identifier) (SR0342.8.7)** process parameters (page 46), and **Boolean property - Master (bundle identifier) (SR0342.8.8)** process parameters (page 47) defines the write sequence of property tag values. Process parameters without property types are skipped. If one of the following issues occurs, phase behavior is as follows:
Tag not enabled	20.1	➢ **Numeric Property Bundle** Phase sets the value to N/A. ➢ **String Property Bundle**, **Boolean Property Bundle** Phase changes cell background to gray.
Not all defined automation properties have a value to be set (value is empty or null)	20.2	■ Phase does not display a value, ■ changes cell background to red, and ■ appends "(X)" to the "empty value". If the **Set** button is used, ■ phase displays **Defined values incomplete (SR0342.3.6.3)** error message (page 54) and ■ does not perform a tag write operation.
Tag write operation fails due to an automation integration issue	20.3	■ Phase does not display a value, ■ does not select read-only checkbox to indicate if the values have been successfully set on the automation layer, ■ changes cell background of checkbox to red, ■ appends "(X)" to the "empty value", and ■ displays **System error (SR0342.3.6.4)** error message (page 54), **Automation error (SR0342.3.6.5)** error message (page 55), or combined error message (**Error message grouping (SR0342.3.6.6)** error message (page 55)).
	30	When the operator has confirmed an error message with **OK**, phase returns to the **Active mode (SR0342.1.2)** layout (page 37). If applicable, continue with the **Override value definition (Numeric property) (SR0342.3.1.2)** user-triggered exception (page 48), **Override value definition (String property) (SR0342.3.1.3)** user-triggered exception (page 50), **Override value definition (Boolean property) (SR0342.3.1.4)** user-triggered exception (page 51), or the **Input at equipment (SR0342.3.1.1)** user-triggered exception (page 48).

Confirm phase (SR0342.2.4)

- Function: Completion of phase
- Type: Main path
- Trigger: Operator confirms phase or **Automatic completion (SR0342.2.2)** mode (page 41) is active
- Postcondition: Phase is completed

Step	#	Description
In **Manual completion (SR0342.2.1)** mode (page 40): Operator confirms phase	10	Operator confirms the tag values.
Phase performs completion checks	20	If one of the following issues occurs, phase displays an error message: ■ Defined property values have not been set, **Defined values not set (SR0342.3.6.7)** error message (page 56). ■ Defined property values have not been set and defined property values are incomplete, combined error message (**Error message grouping - Confirmed (SR0342.3.6.8)** error message (page 56)). When the operator has confirmed an error message with **OK**, phase returns to the **Active mode (SR0342.1.2)** layout (page 37). If applicable, continue with the **Input at equipment (SR0342.3.1.1)** user-triggered exception (page 48).
Phase completion	30	Phase is completed.

Process Parameters (SR0342.8+)

The following process parameters define the behavior of the phase.

BASIC PARAMETERS

Instruction (SR0342.8.1)

➢ For recent changes, see revision history (page 115).

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

Identified equipment entity (SR0342.8.2)

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

Mode (SR0342.8.3)

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

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Override value definition (SR0342.8.4)

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 definition (Numeric property bundle) (SR0342.3.1.2)** user-triggered exception (page 48), **Override value definition (String property bundle) (SR0342.3.1.3)** user-triggered exception (page 50), and **Override value definition (Boolean property bundle) (SR0342.3.21.4)** user-triggered exception (page 51).

Input at equipment (SR0342.8.5)

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 **Input at equipment (SR0342.3.1.1)** user-triggered exception (page 48).

Numeric Property Bundle

Bundle process parameters (Framework capability)

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

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

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A2] (page 113).

Master (Bundle identifier) (SR0342.8.6)

Attribute	Type	Comment
Property	String	Equipment property to be written.
Low	BigDecimal (Double, Float, Integer)	Value to be set.
Value	BigDecimal (Double, Float, Integer)	Value to be set.

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

Property Selection editor (Framework capability)

The system provides a Property Selection editor for selecting an equipment property based on its data type (numeric, string, boolean).

String Property Bundle

Bundle process parameters (Framework capability)

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

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

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A2] (page 113).

Master (Bundle identifier) (SR0342.8.7)

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

Property Selection editor (Framework capability)

The system provides a Property Selection editor for selecting an equipment property based on its data type (numeric, string, boolean).

Boolean Property Bundle

Bundle process parameters (Framework capability)

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

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

This framework capability refers to **Bundle Process Parameters (SR3146.9.7.4.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A2] (page 113).

Master (Bundle identifier) (SR0342.8.8)

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

Property Selection editor (Framework capability)

The system provides a Property Selection editor for selecting an equipment property based on its data type (numeric, string, boolean).

Exceptions (SR0342.3+)

The phase supports user-defined, user-triggered (page 47), system-triggered (page 47), and post-completion exceptions (page 52) and their configuration by means of process parameters (page 43).

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 (SR0342.3.1+)

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.

Input at equipment (SR0342.3.1.1)

The **Input at equipment** exception allows an operator to document that property values have been set manually with an interface connected to the physical equipment. With the exception the operator confirms that the values have been set as documented by this phase.

Representation during exception handling:

- Instruction:
  Values set directly at equipment.
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Input at equipment (SR0342.8.5)** process parameter (page 45))
  Property: <property identifier>
  - Example:
    Value was set manually at the HMI.
    Property: AgitatorSpeed

Input at equipment - Logic (SR0342.3.1.1.1)

- Trigger: Exception is selected
- Postcondition: Value of property is set

Step	#	Description
Operator confirms exception	10	Phase shows exception description to be signed according to **Input at equipment (SR0342.8.5)** process parameter (page 45).
Operator signs exception	20	Phase records exception. Additionally, phase adds the exception marker to each cell in the **Automation set** column of the **Active mode (SR0342.1.2)** layout (page 37).

Numeric Property Bundle

Override value definition (SR0342.3.1.2)

The **Override value definition** exception allows an operator to override the numeric value defined by the process parameter or the entity (see **Numeric property - Master (bundle identifier) (SR0342.8.6)** process parameter (page 45)).

There is one exception per numeric property.
Properties for which an error has been detected are displayed at the top of the list properties.

Representation during exception handling:

- Instruction:
  Override recorded values
  Current values (low, value, high): <current values from process parameter or entity> <UoM>
  Override values (low, value, high): <new values> <UoM>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override value definition (SR0342.8.4)** process parameter (page 44))
  Property: <property identifier>
  Old values (low, value, high):
  <value> <UoM>
  <value> <UoM>
  <value> <UoM>
  New values (low, value, high):
  <value> <UoM>
  <value> <UoM>
  <value> <UoM>
  - Example:
    Values overridden.
    Property: AgitatorSpeed
    Old values (low, value, high):
    12.11 rpm
    12.43 rpm
    13.43 rpm
    New values (low, value, high):
    12.61 rpm
    12.93 rpm
    13.93 rpm

Override value definition - Logic (SR0342.3.1.2.1)

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

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

Step	#	Description
	20	Operator enters values. If the following issue occurs, phase displays an error message: ■ Data format does not match, **Invalid data format error (SR0342.3.6.9)** error message (page 57).
Operator confirms exception	30	If the following issue occurs, phase displays an error message: ■ Override value is missing, **No value overridden (SR0342.3.6.10)** error message (page 57). Phase shows exception description to be signed according to **Override value definition (SR0342.8.4)** process parameter (page 44).
Operator signs exception	40	Phase sets the value, resets the "set on automation layer" indicator, and records the exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0342.1.2)** layout (page 37).

STRING PROPERTY BUNDLE

Override value definition (SR0342.3.1.3)

The **Override value definition** exception allows an operator to override the string value defined by the process parameter or the entity (see **String property - Master (bundle identifier) (SR0342.8.8)** process parameter (page 46)).

There is one exception per string property.
Properties for which an error has been detected are displayed at the top of the list properties.

Representation during exception handling:

- Instruction:
  Override recorded value
  Current value: <current value from process parameter or entity>
  Override value: <value>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Override value definition (SR0342.8.4)** process parameter (page 44))
  Property: <property identifier>
  Old value: <old value>
  New value: <new value>
  - Example:
    Value overridden.

Property: HeatingPerformed
Old value: Temperature alarm
New value: High temperature alarm

Override value definition - Logic (SR0342.3.1.3.1)

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

Step	#	Description
Operator triggers exception	10	Phase displays Exception Window.
	20	Operator enters value.
Operator confirms exception	30	If the following issue occurs, phase displays an error message: ■ Override value is missing, **No value overridden (SR0342.3.6.11)** error message (page 57). Phase shows exception description to be signed according to **Override value definition (SR0342.8.4)** process parameter (page 44).
Operator signs exception	40	Phase sets the value, resets the "set on automation layer" indicator, and records the exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0342.1.2)** layout (page 37).

BOOLEAN PROPERTY BUNDLE

Override value definition (SR0342.3.1.4)

The **Override value definition** exception allows an operator to override the boolean value defined by the process parameter or the entity (see **Boolean property - Master (bundle identifier) (SR0342.8.8)** process parameter (page 47)).

There is one exception per boolean property.
Properties for which an error has been detected are displayed at the top of the list properties.

Representation during exception handling:

- Instruction:
  Override recorded value
  Current value: <current value from process parameter or entity>
  Override value: <available values>
  Confirm button.
- Exception text:
  <Exception text>

(taken from **Override value definition (SR0342.8.4)** process parameter (page 44))
Property: <property identifier>
Old value: <old value>
New value: <new value>

- Example:
  Value overridden.
  Property: Infrared sensor on
  Old value: True
  New value: False

Override value definition - Logic (SR0342.3.1.4.1)

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

Step	#	Description
Operator triggers exception	10	Phase displays Exception Window.
	20	Operator selects value.
Operator confirms exception	30	If the following issue occurs, phase displays an error message: ■ Override value is missing, **No value overridden (SR0342.3.6.12)** error message (page 58). Phase shows exception description to be signed according to **Override value definition (SR0342.8.4)** process parameter (page 44).
Operator signs exception	40	Phase sets the value, resets the "set on automation layer" indicator, and records the exception. Additionally, phase adds the exception marker to the value's cell in the **Active mode (SR0342.1.2)** layout (page 37).

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 (SR0342.3.6+)

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

They are composed of up to three levels:

1. header,
2. category, and
3. details (not always used).

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

Set Property-specific Error Messages (Pre-writing)

Invalid configuration error (SR0342.3.6.1)

UI text	Comment
1. The Set action was not successful. 2. Please set the values directly at the equipment.	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: CheckBeforeExecuteError_HeaderMsg 2. Message pack: PhaseEqmAISetOPCValues<version> Message ID: IrreparableExecution_ErrorCategory Potential error cause: ■ Referenced equipment entity is undefined (Null).

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

Input at equipment recorded (SR0342.3.6.2)

UI text	Comment
1. The Set action was not successful. 2. <empty string> 3. The input at equipment exception has already been recorded.	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: CheckBeforeExecuteError_HeaderMsg 2. Message pack: PhaseEqmAISetOPCValues<version> Message ID: BusinessLogic_ErrorCategory 3. Message pack: PhaseEqmAI<version> Message ID: WritePrecheck_NoSetAfterInputAtEquipment_ErrorMsg Potential error cause: **Set** button is used after the **Input at equipment (SR0342.3.1.1)** user-triggered exception (page 48) has been signed.

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

Defined values incomplete (SR0342.3.6.3)

UI text	Comment
1. The Set action was not successful. 2. Please define the missing values manually.	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: CheckBeforeExecuteError_HeaderMsg 2. Message pack: PhaseEqmAISetOPCValues<version> Message ID: MissingValues_ErrorCategory Potential error cause: One or more values that should be written are not defined.
1. The Set action was not successful. 2. Retry the Set action or set the values directly at the equipment.	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: CheckBeforeExecuteError_HeaderMsg 2. Message pack: PhaseEqmAISetOPCValues<version> Message ID: ReparableExecution_ErrorCategory Potential error cause: Due to a configuration error one or more values could not be written.

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

Set Property-specific Error Messages (Writing)

System error (SR0342.3.6.4)

UI text	Comment
1. The Set action was not successful. 2. <empty string>	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: ExecutionError_HeaderMsg 2. Message pack: PhaseEqmAISetOPCValues<version> Message ID: Other_ErrorCategory Potential error cause: ■ Automation Integration server cannot be reached. ■ Automation Integration server write failure.

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

Automation error (SR0342.3.6.5)

UI text	Comment
1. The Set action was not successful. 2. Please set the values directly at the equipment.	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: ExecutionError_HeaderMsg 2. Message pack: PhaseEqmAISetOPCValues<version> Message ID: IrreparableExecution_ErrorCategory Potential error cause: ■ Tag is enabled but the tag path is undefined. ■ The property to be written is not defined for the identified equipment entity. ■ Live Data server write failure.

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

Error message grouping (SR0342.3.6.6)

If several errors occur during the execution of the Set action, the error messages are combined and displayed in a single error dialog. The three error levels specified in the **Error Messages (SR0342.3.6+)** description (page 53) are used as follows.
The combined error message consists of:

1. One header message,
2. one or more category messages, and
3. one or more detail messages.

UI text	Comment
1. The Set action was not successful. 2. <List of category messages>	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: ExecutionError_HeaderMsg 2. Category of **System error (SR0342.3.6.4)** error message (page 54) and/or **Automation error (SR0342.3.6.5)** error message (page 55). Potential error cause: If there are several errors related to the execution of the **Set** button, the displayed error message contains all error categories and details that apply.

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

Phase Completion-specific Error Messages

Defined values not set (SR0342.3.6.7)

UI text	Comment
1. Cannot confirm 2. Not all values were set successfully at the entity. Retry the Set action or set the values directly at the equipment.	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: CompletionError_HeaderMsg 2. Message pack: PhaseEqmAISetOPCValues<version> Message ID: ReparableConfirm_ErrorCategory Potential error cause: ■ One or more values that should be written are not defined. ■ One or more values that should be written cannot be set on the automation layer.

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

Error message grouping - Confirm (SR0342.3.6.8)

If several errors occur when the phase is confirmed, the error messages are combined and displayed in a single error dialog. The three error levels specified in the **Error Messages (SR0342.3.6+)** description (page 53) are used as follows.
The combined error message consists of:

1. One header message,
2. one or more category messages, and
3. one or more detail messages.

UI text	Comment
1. Cannot confirm 2. <List of category messages>	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: CompletionError_HeaderMsg 2. Category of **System error (SR0342.3.6.4)** error message (page 54) and/or **Automation error (SR0342.3.6.5)** error message (page 55). Potential error cause: If there are several errors related to phase completion, the displayed error message contains all error categories and details that apply.

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

User-triggered Exception-specific Error Messages

Invalid data format error (SR0342.3.6.9)

UI text	Comment
1. Cannot confirm the overridden values. 2. <empty string> 3. The value entered for <attribute name> has an unsuitable format. Please enter a value that is valid for the <data type name> data type.	1. Message pack: PhaseEqmAISetOPCValues<version> Message ID: OverrideExceptionConfirmationError_HeaderMsg 2. Message pack: PhaseEqmAISetOPCValues<version> Message ID: Other_ErrorCategory 3. Message pack: PhaseEqmAI<version> Message ID: OverrideInvalidDataFormat_ErrorMsg Potential error cause: The entered text value cannot be converted to a numeric value of the targeted numeric data type.

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

NUMERIC PROPERTY BUNDLE

No value overridden (SR0342.3.6.10)

UI text	Comment
1. Enter an override value.	1. Message pack: PhaseEqmAI<version> Message ID: OverrideNumericValueNotSet_ErrorMsg Potential error cause: No override value was entered before the user-triggered exception was confirmed.

STRING PROPERTY BUNDLE

No value overridden (SR0342.3.6.11)

UI text	Comment
1. You have to enter an override value before you can confirm.	1. Message pack: PhaseEqmAI<version> Message ID: OverrideStringValueNotSet_ErrorMsg Potential error cause: No override value was entered before the user-triggered exception was confirmed.

BOOLEAN PROPERTY BUNDLE

No value overridden (SR0342.3.6.12)

UI text	Comment
1. You have to select a value before you can confirm.	1. Message pack: PhaseEqmAI<version> Message ID: OverrideBooleanValueNotSet_ErrorMsg Potential error cause: No override value was entered before the user-triggered exception was confirmed.

Output Variables (SR0342.9+)

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.

Automation set successful (SR0342.9.1)

- Data type: Boolean
- Values: `true`, `false`

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

Numeric Property 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 113).

Value (SR0342.9.2)

- Data type: BigDecimal
- Usage: The output variable provides the actual value of the numeric property tag as a **BigDecimal** value. The value is Null if N/A is the phase result.

Low (SR0342.9.3)

- Data type: BigDecimal
- Usage: The output variable provides the value of the numeric property tag as a **BigDecimal** value. The value is Null if N/A is the phase result.

High (SR0342.9.4)

- Data type: BigDecimal
- Usage: The output variable provides the value of the numeric property tag as a **BigDecimal** value. The value is Null if N/A is the phase result.

Unit of measure (SR0342.9.5)

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

Automation set successful (SR0342.9.6)

- Data type: Boolean
- Values: `true`, `false`

- Usage: The output variable states if the set operation on the automation layer was successful.
  - The value is `true` if all property values of the numeric property tag have been set.
  - The value is `false` if at least one of the property values of the numeric property tag could not be set.

String Property 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 113).

Value (SR0342.9.7)

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

Automation set successful (SR0342.9.8)

- Data type: Boolean
- Values: `true`, `false`
- Usage: The output variable states if the set operation on the automation layer was successful.
  - The value is `true` if the property value of the string property tag has been set.
  - The value is `false` if the property values of the string property tag could not be set.

Boolean Property 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 113).

Value (SR0342.9.9)

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

Automation set successful (SR0342.9.10)

- Data type: Boolean
- Values: `true`, `false`
- Usage: The output variable states if the set operation on the automation layer was successful.
  - The value is `true` if the property value of the boolean property tag has been set.
  - The value is `false` if the property values of the boolean property tag could not be set.

Performance (SR0342.12+)

Performance of Set Activity (SR0342.12.1)

The time for setting the OPC values on the automation layer does not take longer than 5 seconds. Any potential delay by the OPC server or the PLC communication is not considered.

Chapter 4

Monitor Numeric Value Phase (SR0360+)

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

An example use case is:

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

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

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

The affected equipment entity, monitoring period, condition string, and the timestamp when the condition is met are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 66).

The condition will be considered to be met if the value matches its requirements and the corresponding tag quality is **Good**.

Both the monitoring period and the tag update rate are configurable; the tag update rate in Data Manager on equipment property level, the monitoring period in Recipe and Workflow Designer on process parameter level.

TIP

- Changes of a value that fulfill the condition only temporarily between two read-cycles are not detected.
- Due to technical reasons, the accuracy of the **Double** or **Float** numeric data type cannot be guaranteed. Hence, the result of the **Value == 13.4** condition can be unexpected if the value is of the **Double** or **Float** numeric data type.

Anomalies that occur during processing are covered by the phase exception handling (page 71) (e.g. condition not fulfilled within monitor period).

After completion the phase displays the affected equipment entity, condition string, monitor duration and the phase result in the Execution Window.

The Navigator displays the identifier of the affected equipment entity.

Figure 3: Monitor numeric value during execution

Layout

The phase provides individual layouts for its representation during execution (page 64), in the Navigator (page 65), and in the sub-report (page 66).

Representation during Execution (SR0360.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0360.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0360.8.1)** process parameter (page 68))
2. Equipment entity:
   Condition: [<value> <comparator>] <property identifier> <comparator> <value> (<meaning of "condition met">)
   Duration: <d hh mm ss>
   (taken from **Numeric property (SR0360.8.4)** process parameter (page 69))
   Result:
   Value:
3. **Confirm** button (disabled).

Active mode (SR0360.1.2)

1. <Instruction text>
   (taken from **Instruction (SR0360.8.1)** process parameter (page 68))
2. Equipment entity: <equipment entity identifier>/<equipment entity short description>

(taken from **Identified equipment entity (SR0360.8.2)** process parameter (page 68))
Condition: [<value> <comparator>] <property identifier> <comparator> <value> (<meaning of "condition met">)
Duration: <d hh mm ss> (until end time <end time>)
(taken from **Numeric property (SR0360.8.4)** process parameter (page 69))
Result: **Condition met** or **Condition not met**
Value: <value> <UoM> (<tag timestamp> or **N/A**)

- Tag timestamp: e.g. 02/22/2013 12:49:09 PM CET
- Manual in case a value has been overridden by using the **Stop monitoring and record result (SR0360.3.1.1)** user-triggered exception (page 73).

3. For the representation, see **Monitor a numeric value (SR0360.2.1)** function (page 67).
4. **Confirm** button.

Completed mode (SR0360.1.3)

1. <Instruction text>
(taken from **Instruction (SR0360.8.1)** process parameter (page 68))
2. Equipment entity: <equipment entity identifier>/<equipment entity short description>
(taken from **Identified equipment entity (SR0360.8.2)** process parameter (page 68))
Condition: [<value> <comparator>] <property identifier> <comparator> <value> (<meaning of "condition met">)
Duration: <d hh mm ss>
(taken from **Numeric property (SR0360.8.4)** process parameter (page 69))
Result: **Condition met** or **Condition not met**
Value: <value> <UoM> (<tag timestamp> or **N/A**)

   - Tag timestamp: e.g. 02/22/2013 12:49:09 PM CET
   - Manual in case a value has been overridden by using the **Stop monitoring and record result (SR0360.3.1.1)** user-triggered exception (page 73).

3. For the representation, see **Monitor a numeric value (SR0360.2.1)** function (page 67).
4. **Confirm** button (completed).

Representation in Navigator (SR0360.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>
  - Example:
    Monitor mixer value

Information column (SR0360.4.1)

- <Identifier of affected equipment entity>
  - Example: MixerA12

Action column

- There are no actions available.

Representation in Sub-report (SR0360.5+)

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 (SR0360.5.1)

- Instruction text
- Entity (identifier and short description)
- Duration
- Condition string ([<value>] <comparator> <value> <meaning of "condition met">)
- Result
- Value
- Timestamp of tag value

Business Logic (SR0360.2+)

The phase implements the following business logic.

Monitor a numeric value (SR0360.2.1)

- Function: Monitor a numeric value of affected equipment entity and property
- Trigger: Phase becomes active
- Postcondition: Monitoring is completed

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0360.1.2)** layout (page 64). If monitoring could not be activated, phase displays **Automation error (SR0360.3.6.1)** error message (page 75).
Phase monitors value	20	Within the defined period (**Numeric property (SR0360.8.4)** process parameter (page 69)), phase monitors the numeric value and updates phase representation. As long as the value is monitored, phase displays end time to the right of the duration. Monitoring is terminated when ■ condition is fulfilled (see step 30) or ■ monitoring duration has expired (see step 40).
Phase behavior when condition is fulfilled	30	Values are only evaluated against the pre-defined condition if the tag quality is **Good**. Bad quality tags are not evaluated. ■ Phase displays the result as **Condition met**. Phase displays the read value and its timestamp. ■ If **Mode (SR0360.8.3)** process parameter (page 85) is set to **Automatic completion** and **Meaning of "condition met"** attribute of **Numeric property (SR0360.8.4)** process parameter (page 69) is set to **Good**, phase is completed. ■ If **Meaning of "condition met"** of **Numeric property (SR0360.8.4)** process parameter (page 69) is set to **Exception**: Phase creates **Monitoring exception (SR0360.3.2.1)** system-triggered exception (page 72) and must be completed manually.
Phase behavior when monitoring duration has expired	40	■ Phase displays the result as **Condition not met**. Phase displays the values as **N/A** with **N/A** as timestamp. ■ If **Meaning of "condition met"** attribute of **Numeric property (SR0360.8.4)** process parameter (page 69) is set to **Good**: Phase creates **Monitoring exception (SR0360.3.2.1)** system-triggered exception (page 72) and must be completed manually. ■ If **Mode (SR0360.8.3)** process parameter (page 85) is set to **Automatic completion** and **Meaning of "condition met"** attribute of **Numeric**

Step	#	Description
		property (SR0360.8.4) process parameter (page 69) is set to **Exception**, phase is completed.
Operator confirms phase	50	■ If monitoring is active, phase displays **Monitoring in progress (SR0360.3.6.2)** error message (page 76). In this case, the **Stop monitoring and record result (SR0360.3.1.1)** user-triggered exception (page 73) needs to be recorded before the phase can be completed. ■ If a system-triggered exception has been raised before, it needs to be recorded before the phase can be completed. ■ Otherwise, phase is completed.

Process Parameters (SR0360.8+)

The following process parameters define the behavior of the phase.

BASIC PARAMETERS

Instruction (SR0360.8.1)

➢ For recent changes, see revision history (page 115).

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	

Identified equipment entity (SR0360.8.2)

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

Mode (SR0360.8.3)

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

PROPERTY TYPE PARAMETERS

Numeric property (SR0360.8.4)

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

Attribute	Type	Comment
		configured on equipment property level.

Property Selection editor (Framework capability)

The system provides a Property Selection editor for selecting an equipment property based on its data type (numeric, string, boolean).

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Monitoring exception (SR0360.8.5)

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 **Monitoring exception (SR0360.3.2.1)** system-triggered exception (page 72).

Unforeseen resume (SR0360.8.7)

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**.

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 **Unforeseen resume (SR0360.3.2.2)** system-triggered exception (page 72).

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Stop monitoring and record result (SR0360.8.6)

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 **Stop monitoring and record result (SR0360.3.1.1)** user-triggered exception (page 73).

Exceptions (SR0360.3+)

The phase supports user-defined, user-triggered (page 73), system-triggered (page 72), and post-completion exceptions (page 75) and their configuration by means of process parameters (page 68).

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

System-triggered Exceptions (SR0360.3.2+)

➢ For recent changes, see revision history (page 115).

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.

Monitoring exception (SR0360.3.2.1)

The exception text is extended by messages specific to the current situation.

Representation of the exception:

- <Exception text>
  (taken from **Monitoring exception (SR0360.8.5)** process parameter (page 70))
  - Condition <property identifier> <comparator> <value> (<meaning of exception>) met by [<monitoring result value> | N/A]
  - Condition <property identifier> <comparator> <value> (<meaning of exception>) not met by [<monitoring result value> | N/A]
  - Example:
    Monitoring exception occurred:
    Condition Speed >= 50 rpm (Good) met by 51.4 rpm

Monitoring exception - Logic (SR0360.3.2.1.1)

- Trigger: Either the condition is fulfilled and defined as **Exception** or the condition is defined as **Good** and the monitor duration has expired.
  For the settings, see **Comparator**, **Meaning of "condition met"**, and **Monitor duration** attributes of the **Numeric property (SR0360.8.4)** process parameter (page 69).
- Postcondition: Exception is recorded

Step	#	Description
Operator triggers exception	10	Phase records exception.
	20	Phase returns to **Active mode (SR0360.1.2)** layout (page 64).

Unforeseen resume (SR0360.3.2.2)

Representation of the exception:

- <Exception text>
  (taken from **Unforeseen resume (SR0360.8.7)** process parameter (page 70))

The system has been resumed while monitoring a numeric value of <property identifier>. It must be ensured that the data recorded so far matches the physical situation on the shop floor.
Consider to stop monitoring and record an exception with the monitoring result.

- Example:
  A critical resume situation has occurred. Contact your supervisor before proceeding.
  The system has been resumed while monitoring a numeric value of speed. It must be ensured that the data recorded so far matches the physical situation on the shop floor.
  Consider to stop monitoring and record an exception with the monitoring result.

Unforeseen resume - Logic (SR0360.3.2.2.1)

- Trigger: Monitoring a numeric value has been interrupted while the phase was active so that the system needs to be resumed
- Postcondition: Phase is back in active mode

Step	#	Description
Phase activation	10	Phase displays the **Unforeseen resume (SR0360.3.2.2)** system-triggered exception.
Operator triggers exception	20	Phase records exception.
	30	Phase restarts monitoring again with the full monitor duration configured (taken from **Numeric property (SR0360.8.4)** process parameter (page 69)).
	40	Phase returns to **Active mode (SR0365.1.2)** layout (page 80).

User-triggered Exceptions (SR0360.3.1+)

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.

Stop monitoring and record result (SR0360.3.1.1)

The **Stop monitoring and record result** exception allows an operator to terminate monitoring before the monitoring duration has expired and to record the monitoring result value manually.

The exception is only enabled if monitoring is active.

Representation during exception handling:

- Instruction:
  Stop and record the monitoring value of <property identifier>
  Value: <value><UoM>
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Stop monitoring and record result (SR0360.8.6)** process parameter (page 71))
  Property: <property identifier>
  Recorded value: <value> <UoM>
  - Example:
    Monitoring stopped manually.
    Property: temperature
    Recorded value: 33 °C

Stop monitoring - Logic (SR0360.3.1.1.1)

- Trigger: Exception is selected
- Postcondition: Monitoring of numeric value is stopped and value is recorded

Step	#	Description
Operator triggers exception	10	Phase shows exception description to be signed according to **Stop monitoring and record result (SR0360.8.6)** process parameter (page 71).
Operator provides value	20	If the following issue occurs, phase displays an error message: ■ Value is entered and data format does not match, **Invalid data format error (SR0360.3.6.3)** error message (page 76).
Operator signs exception	30	Phase records exception.
Phase activation	40	Phase returns to the **Active mode (SR0360.1.2)** layout (page 64).
Phase behavior when a value is provided	50	Phase evaluates the condition and displays the result correspondingly: **Condition met** or **Condition not met**. ■ If **Meaning of "condition met"** attribute of **Numeric property (SR0360.8.4)** process parameter (page 69) is set to **Good**: Phase displays the entered value and **N/A** as timestamp. If **Mode (SR0360.8.3)** process parameter (page 85) is set to **Automatic completion**, phase is completed. ■ If **Meaning of "condition met"** of **Numeric property (SR0360.8.4)** process parameter (page 69) is set to **Exception**: Phase displays the entered value and **Manual** as timestamp.

Step	#	Description
		Phase creates **Monitoring exception (SR0360.3.2.1)** system-triggered exception (page 72) and must be completed manually.

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 (SR0360.3.6+)

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.

Automation error (SR0360.3.6.1)

UI text	Comment
Cannot monitor the value of the <property identifier> property. Please verify your system configuration and try again or override. System errors: <automation-related message>.	Message pack: EQAIMonNumeric<version> Message ID: AutomationErrorMsg

Monitoring in progress (SR0360.3.6.2)

UI text	Comment
Cannot confirm, since monitoring is in progress.	Message pack: EQAIMonNumeric<version> Message ID: CannotComplete_ErrorMsg

Invalid data format error (SR0360.3.6.3)

UI text	Comment
1. Cannot confirm the overridden values. 2. <empty string> 3. The value entered for <attribute name> has an unsuitable format. Please enter a value that is valid for the <data type name> data type.	1. Message pack: PhaseEqmAISetManager<version> Message ID: OverrideExceptionConfirmationError_HeaderMsg 2. Message pack: PhaseEqmAISetManager<version> Message ID: Other_ErrorCategory 3. Message pack: PhaseEqmAI<version> Message ID: OverrideInvalidDataFormat_ErrorMsg Potential error cause: The entered text value cannot be converted to a numeric value of the targeted numeric data type.

Output Variables (SR0360.9+)

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.

Monitoring exception occurred (SR0360.9.1)

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

Value (SR0360.9.3)

- Data type: MeasuredValue
- Usage: The output variable provides the value and its unit of measure as a **MeasuredValue** object. The value is Null if N/A is the phase result.

Timestamp of tag (SR0360.9.2)

- Data type: Timestamp
- Usage: The output variable provides the timestamp when the value was read.

Get Alarms Phase (SR0365+)

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

An example use case is:

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

The phase supports up to 20 alarm tags.
The tag quality is not evaluated while determining if an alarm or error condition has been reached.

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

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

The affected equipment entity, the affected alarm property, its alarm identifier, statuses, and timestamps are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 82).

Anomalies that occur during processing are covered by the phase exception handling (page 86) (e.g. tag cannot be read).

The list of occurred alarms is available as phase output. Depending on the recipe design, this list of alarms can be evaluated within a transition condition in order to control related follow-up production steps.

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

The Navigator displays the identifier of the affected equipment entity.

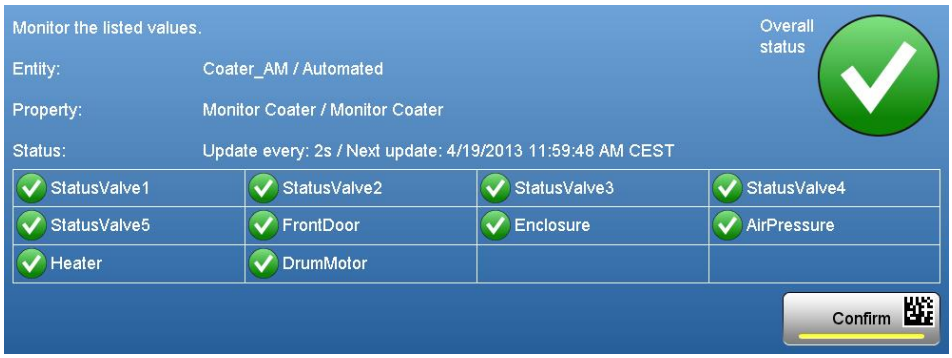

Figure 4: Get alarms during execution

Layout

The phase provides individual layouts for its representation during execution (page 80), in the Navigator (page 82), and in the sub-report (page 82).

Representation during Execution (SR0365.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0365.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0365.8.1)** process parameter (page 84))
2. Entity:
   Property: <property identifier> / <property short description>
   (taken from **Alarm property (SR0365.8.4)** process parameter (page 85))
3. Status: N/A
4. List of alarms (with status and identifier)
   (taken from **Alarm property (SR0365.8.4)** process parameter (page 85))
5. **Confirm** button (disabled).

Active mode (SR0365.1.2)

1. <Instruction text>
   (taken from **Instruction (SR0365.8.1)** process parameter (page 84))
2. Entity: <equipment entity identifier>/<equipment entity short description>
   (taken from **Identified equipment entity (SR0365.8.2)** process parameter (page 84))
   Property: <property identifier> / <property short description>
   (taken from **Alarm property (SR0365.8.4)** process parameter (page 85))

3. Status: <status of tag reading>
   - If reading has been stopped: Stopped at <timestamp>
   - If reading is ongoing: Update every: <hh:mm:ss> / Next update: <timestamp>
     (taken from **Alarm property (SR0365.8.4)** process parameter (page 85))
4. Overall status <icon>
   - Checkmark, if no alarm has been raised, no error has occurred, and all tags have been read successfully.
   - Alarm symbol, if at least one alarm has been raised, regardless of the statuses of all other tags.
   - Error symbol, if at least one error has occurred and no alarms have been raised, regardless of whether all other alarm tags have been read successfully.
5. List of alarms (with status and identifier)
   (taken from **Alarm property (SR0365.8.4)** process parameter (page 85))
   - Checkmark
   - Alarm symbol
   - Error symbol
6. **Confirm** button.

Completed mode (SR0365.1.3)

1. <Instruction text>
   (taken from **Instruction (SR0365.8.1)** process parameter (page 84))
2. Entity: <equipment entity identifier>/<equipment entity short description>
   (taken from **Identified equipment entity (SR0365.8.2)** process parameter (page 84))
   Property: <property identifier> / <property short description>
   (taken from **Alarm property (SR0365.8.4)** process parameter (page 85))
3. Status: Stopped at <timestamp>
4. Overall status <icon>
   - Checkmark, if no alarm has been raised, no error has occurred, and all tags have been read successfully.
   - Alarm symbol, if at least one alarm has been raised, regardless of the statuses of all other tags.
   - Error symbol, if at least one error has occurred and no alarms have been raised, regardless of whether all other alarm tags have been read successfully.

5. List of alarms (with status and identifier)
   (taken from **Alarm property (SR0365.8.4)** process parameter (page 85))
   - Checkmark
   - Alarm symbol
   - Error symbol
6. **Confirm** button (completed).

Representation in Navigator (SR0365.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>
  - Example:
    Get mixer alarms

Information column (SR0365.4.1)

- <Identifier of affected equipment entity>
  - Example: MixerA12

Action column

- There are no actions available.

Representation in Sub-report (SR0365.5+)

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 (SR0365.5.1)

- Instruction text
- Entity (identifier and short description)
- Alarm property (identifier and short description)

- Overall status
- List of alarms
  - Identifier
  - Status (Good, Alarm, Error)
  - Timestamp

Business Logic (SR0365.2+)

The phase implements the following business logic.

Check for alarms (SR0365.2.1)

- Function: Check for alarms of affected equipment entity and property
- Trigger: Phase becomes active
- Postcondition: Check for alarms is completed

Step	#	Description
Phase activation	10	Phase displays its user interface according to the **Active mode (SR0365.1.2)** layout (page 80). If the alarm check cannot be started, phase displays **Invalid property configuration error (SR0365.3.6.2)** error message (page 89) or **Invalid entity configuration error (SR0365.3.6.3)** error message (page 90).
Phase polls tags	20	Within the defined interval (**Alarm property (SR0365.8.4)** process parameter (page 85)), phase polls alarms tags for their status and updates phase representation. Polling is terminated when ■ at least one alarm is raised (see step 30), ■ at least one error has occurred, or ■ the operator confirms the phase (see step 50).
Phase behavior when at least one alarm or one error is raised	30	■ **Enabled** attribute of **Alarm exception (SR0365.8.5)** process parameter (page 85) is set to **Yes**: Phase creates **Alarm exception (SR0365.3.2.1)** system-triggered exception (page 87) and must be completed manually. ■ **Enabled** attribute of **Alarm exception (SR0365.8.5)** process parameter (page 85) is set to **No**: Phase checks setting of **Mode (SR0365.8.3)** process parameter (page 85) (see step 40).

Step	#	Description
Phase checks setting of **Mode (SR0365.8.3)** process parameter (page 85)	40	■ Mode is set to **Manual completion**: Phase stops checking for alarms. ■ Mode is set to **Automatic completion**: Phase stops checking for alarms and is completed.
Operator confirms phase	50	■ If value retrieval is in progress, phase cannot perform the **Confirm** action and displays **Value retrieval in progress (SR0365.3.6.1)** error message (page 89). ■ Phase checks for not yet recorded **Alarm exception (SR0365.3.2.1)** system-triggered exceptions (page 87). If no exception has been recorded, at least one alarm is raised, and **Enabled** attribute of **Alarm exception (SR0365.8.5)** process parameter (page 85) is set to **Yes**, phase creates **Alarm exception (SR0365.3.2.1)** system-triggered exception (page 87) again. ■ Phase is completed.

Process Parameters (SR0365.8+)

The following process parameters define the behavior of the phase.

BASIC PARAMETERS

Instruction (SR0365.8.1)

➢ For recent changes, see revision history (page 115).

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	

Identified equipment entity (SR0365.8.2)

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

Mode (SR0365.8.3)

Attribute	Type	Comment
Mode	Choice list	Defines the processing mode. **Manual completion** (default): Operator confirms the phase. **Automatic completion**: Phase automatically gets the property value and is completed. In this mode, avoid infinite looping by properly resetting alarms before re-entering the phase.

PROPERTY TYPE PARAMETERS

Alarm property (SR0365.8.4)

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

Property Selection editor (Framework capability)

The system provides a Property Selection editor for selecting an equipment property based on its data type (numeric, string, boolean).

CONFIGURATION OF SYSTEM-TRIGGERED EXCEPTIONS

Alarm exception (SR0365.8.5)

Attribute	Type	Comment
Enabled	Flag	Controls if a check is performed. If so, the phase creates a system-triggered exception when an alarm is raised. Default setting: **Yes**.

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

See also **Alarm exception (SR0365.3.2.1)** system-triggered exception (page 87).

Unforeseen resume (SR0365.8.6)

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 **Unforeseen resume (SR0365.3.2.2)** system-triggered exception (page 88).

Exceptions (SR0365.3+)

The phase supports user-defined, user-triggered (page 88), system-triggered (page 87), and post-completion exceptions (page 88) and their configuration by means of process parameters (page 84).

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

System-triggered Exceptions (SR0365.3.2+)

➢ For recent changes, see revision history (page 115).

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.

Alarm exception (SR0365.3.2.1)

Representation of the exception:

- <Exception text>
  (taken from **Alarm exception (SR0365.8.5)** process parameter (page 85))
  The exception text is extended by additional information specific to the exception case:
  - Alarms and errors:
    Received: <alarm identifier 1>; <alarm identifier 2>
    Not readable: <alarm identifier 3, non-boolean>;<alarm identifier 4, non-boolean>
  - Alarms:
    Received: <alarm identifier 1>; <alarm identifier 2>
  - Errors:
    Not readable: <alarm identifier 3, non-boolean>;<alarm identifier 4, non-boolean>
  - Example:
    Alarm exception occurred:
    Received: AlarmTagSensor1; AlarmTagLidOpen13
    Not readable: AlarmTagVessels17

Alarm exception - Logic (SR0365.3.2.1.1)

- Trigger: At least one alarm has been raised or one error has occurred
- Postcondition: Exception is recorded

Step	#	Description
Operator triggers exception	10	Phase records exception.
	20	Phase returns to **Active mode (SR0365.1.2)** layout (page 80).

Unforeseen resume (SR0365.3.2.2)

Representation of the exception:

- <Exception text>
  (taken from **Unforeseen resume (SR0365.8.6)** process parameter (page 86))
  The system has been resumed while monitoring <property identifier>. It must be ensured that the data recorded so far matches the physical situation on the shop floor.
  Consider to stop monitoring.
  - Example:
    A critical resume situation has occurred. Contact your supervisor before proceeding.
    The system has been resumed while monitoring AlarmsTagsMixer. It must be ensured that the data recorded so far matches the physical situation on the shop floor.
    Consider to stop monitoring.

Unforeseen resume - Logic (SR0365.3.2.2.1)

- Trigger: Monitoring alarms has been interrupted while the phase was active so that the system needs to be resumed
- Postcondition: Phase is back in active mode and alarm monitoring is restarted

Step	#	Description
Phase activation	10	Phase displays the **Unforeseen resume (SR0365.3.2.2)** system-triggered exception.
Operator triggers exception	20	Phase records exception.
	30	Phase returns to **Active mode (SR0365.1.2)** layout (page 80).
	40	Phase restarts alarm monitoring.

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 (SR0365.3.6+)

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

They are composed of up to three levels:

1. header,
2. category, and
3. details (not always used).

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

Value retrieval in progress (SR0365.3.6.1)

UI text	Comment
1. Cannot confirm. Please wait while the values are being retrieved.	1. Message pack: PhaseEqmAIGetAlarms<version> Message ID: alarmStatusNotSet_ErrorMsg

Invalid property configuration error (SR0365.3.6.2)

UI text	Comment
1. Cannot start the alert monitoring. 2. <empty string> 3. The <entity identifier> entity does not have a <property identifier> property.	1. Message pack: PhaseEqmAIGetAlarms<version> Message ID: CheckBeforeExecuteError_HeaderMsg 2. Message pack: PhaseEqmAIGetAlarms<version> Message ID: IrreparableExecution_ErrorCategory 3. Message pack: PhaseEqmAI<version> Message ID: PropertyNotFulfilled_ErrorMsg Potential error cause: ■ The property to be read is not defined for the identified equipment entity.

Invalid entity configuration error (SR0365.3.6.3)

UI text	Comment
1. Cannot start the alert monitoring. 2. <empty string> 3. Cannot find the expected entity.	1. Message pack: PhaseEqmAIGetAlarms<version> Message ID: CheckBeforeExecuteError_HeaderMsg 2. Message pack: PhaseEqmAIGetAlarms<version> Message ID: IrreparableExecution_ErrorCategory 3. Message pack: PhaseEqmAI<version> Message ID: IdentifiedEquipmentNull_ErrorMsg Potential error cause: ■ The equipment entity to be used cannot be found. Potentially the recipe parameter is not defined.

Output Variables (SR0365.9+)

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.

Alarm tags (SR0365.9.1)

- Data type: String
- Usage: The output variable provides a semicolon-separated list of the tag identifiers for which an alarm has been set. The list is empty if no alarm has occurred.

Overall status (SR0365.9.2)

- Data type: String
- Values: `GOOD`, `ALARM`, `ERROR`
- Usage: The output variable provides the overall status of the alarm tags.
  - The value is `GOOD` if no alarm has been raised, no error has occurred, and all tags have been read successfully.
  - The value is `ALARM` if at least one alarm has been raised, regardless of the statuses of all other tags.
  - The value is `ERROR` if at least one error has occurred and no alarms have been raised, regardless of whether all other alarm tags have been read successfully.

Chapter 6

Show Historical Data Chart Phase (SR0110+)

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

Example use cases are:

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

The phase supports FactoryTalk Historian.

TIP

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

Values are only rendered in the chart when they exist at the specific point of time and lie within the configured period and y-axis values of the plot.

The affected equipment entity, the query template, and the time series chart are stored in the batch record, thereby becoming available for documentation purposes in the sub-report and batch report (page 95).
Anomalies that occur during processing are covered by the phase exception handling (page 103) (e.g. no chart available).

After completion the phase displays the affected equipment entity and the time series chart in the Execution Window.

The Navigator displays the identifier of the affected equipment entity.

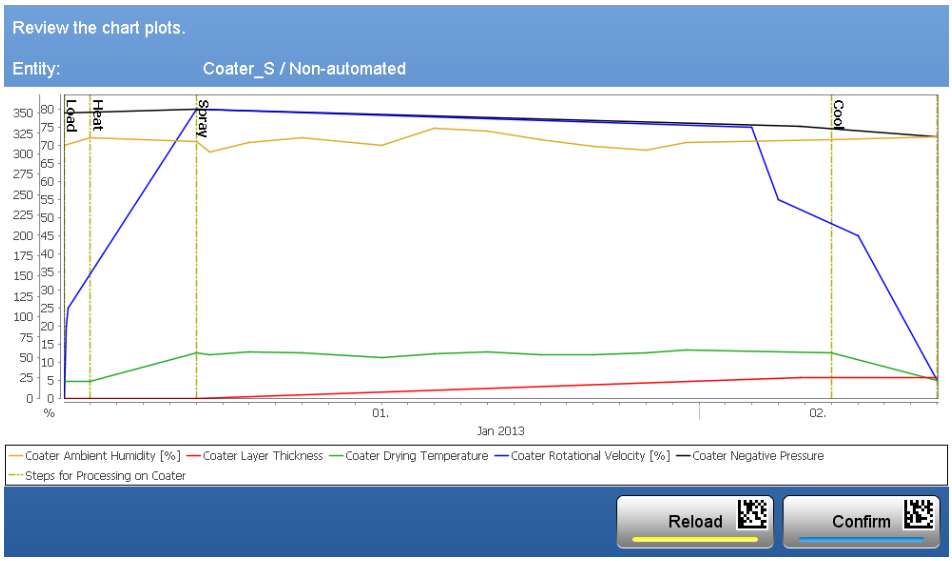

Figure 5: Show historical data chart during execution

Layout

The phase provides individual layouts for its representation during execution (page 94), in the Navigator (page 95), and in the sub-report (page 95).

Representation during Execution (SR0110.1+)

The representation during execution depends on the phase mode.

Preview mode (SR0110.1.1)

1. <Instruction text>
   (taken from **Instruction (SR0110.8.1)** process parameter (page 99))
2. Entity:
3. **Reload** button (disabled).
4. **Confirm** button (disabled).

Active mode (SR0110.1.2)

1. <Instruction text>
   (taken from **Instruction (SR0110.8.1)** process parameter (page 99))
2. Entity: <equipment entity identifier>
   (taken from **Identified equipment entity (SR0110.8.2)** process parameter (page 100))

3. Time series chart
   (configuration taken from **Chart plot (SR0110.8.3)** process parameter (page 100) and **Chart axis (SR0110.8.4)** process parameter (page 101))
4. **Reload** button.
5. **Confirm** button.

Completed mode (SR0110.1.3)

1. <Instruction text>
   (taken from **Instruction (SR0110.8.1)** process parameter (page 99))
2. Entity: <equipment entity identifier>
   (taken from **Identified equipment entity (SR0110.8.2)** process parameter (page 100))
3. Time series chart
   (configuration taken from **Chart plot (SR0110.8.3)** process parameter (page 100) and **Chart axis (SR0110.8.4)** process parameter (page 101))
4. **Reload** button (disabled).
5. **Confirm** button (completed).

Representation in Navigator (SR0110.4+)

The Navigator provides the following details:

Phase column (Framework capability)

- <Phase name>
  - Example:
    Review Data Chart

Information column (SR0110.4.1)

- <Identifier of affected equipment entity>
  - Example: Coater_S

Action column

- There are no actions available.

Representation in Sub-report (SR0110.5+)

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 (SR0110.5.1)

- Time series chart
- Instruction text
- Entity (identifier)
- Queries
  - Property (identifier)
  - Query template (identifier)
  - Executed query (symbolic notation with input values represented as string; not necessarily directly executable due to potentially required data type conversion functions when executed on SQL layer)

Business Logic (SR0110.2+)

The phase implements the following business logic.

Display chart (SR0110.2.1)

- Function: Display a chart with values from the Historian system
- Trigger: Phase becomes active
- Postcondition: Chart is displayed

Step	#	Description
Check configuration	10	Phase displays its user interface according to the **Active mode (SR0110.1.2)** layout (page 94) with the chart image placeholder and the embedded text **No chart generated**. Phase evaluates the configuration of the **Identified equipment entity (SR0110.8.2)** process parameter (page 100), **Chart plot (SR0110.8.3)** process parameter (page 100), and **Chart axis (SR0110.8.4)** process parameter (page 101) for consistency. If one of the following issues occurs, phase displays the **Invalid configuration error (SR0110.3.6.1)** error message (page 104):

Step	#	Description
		■ Identified equipment entity is not defined. ■ Configured chart plot property is not defined for the entity. ■ No Y-axis with a corresponding unit of measure is defined for chart plot property. ■ There are more properties with different units of measure than corresponding y-axes. ■ There are several Y-axes defined with the same unit of measure (duplicate). ■ No query template has been selected for an enabled chart plot. ■ No property has been selected for an enabled chart plot. ■ No plot format has been selected for an enabled chart plot. ■ No Y-axes range values have been defined if **Autorange** is disabled for an enabled chart axis.
Data retrieval	20	Phase retrieves data from the Historian system based on the parameterized query template. Phase displays chart image placeholder with loading data indicator while retrieving data. If the data cannot be loaded, phase displays the **Data retrieval error (SR0110.3.6.2)** error message (page 106).
Display chart	30	■ If data retrieval was successful, phase saves the generated chart image and displays it. ■ Otherwise, phase displays chart image placeholder with the embedded text **No chart generated**.

Reload chart (SR0110.2.2)

- Function: Execute data retrieval and update the displayed chart
- Trigger: Operator reloads chart data
- Postcondition: Reloaded chart is displayed

Step	#	Description
Check configuration	10	Phase evaluates the configuration of the **Identified equipment entity (SR0110.8.2)** process parameter (page 100), **Chart plot (SR0110.8.3)** process parameter (page 100), and **Chart axis (SR0110.8.4)** process parameter (page 101) for consistency.

Step	#	Description
		If one of the following issues occurs, phase displays the **Invalid configuration error (SR0110.3.6.1)** error message (page 104): ■ Identified equipment entity is not defined. ■ Configured chart plot property is not defined for the entity. ■ No Y-axis with a corresponding unit of measure is defined for chart plot property. ■ There are more properties with different units of measure than corresponding y-axes. ■ There are several Y-axes defined with the same unit of measure (duplicate). ■ No query template has been selected for an enabled chart plot. ■ No property has been selected for an enabled chart plot. ■ No plot format has been selected for an enabled chart plot. ■ No Y-axes range values have been defined if **Autorange** is disabled for an enabled chart axis.
Data retrieval	20	Phase retrieves data from the Historian system based on the parameterized query template. Phase displays chart image placeholder with loading data indicator while retrieving data. If the data cannot be loaded, phase displays the **Data retrieval error (SR0110.3.6.2)** error message (page 106).
Display chart	30	■ If data retrieval was successful, phase saves the generated chart image and displays it. ■ Otherwise, phase displays chart image placeholder with the embedded text **No chart generated**.

Confirm phase (SR0110.2.3)

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

Step	#	Description
Operator confirms phase	10	■ If no chart is displayed and the **Chart unavailable (SR0110.3.1.1)** user-triggered exception (page 103) has not been signed, phase displays **Chart unavailable error (SR0110.3.6.2)** error message (page 106). In this case, the **Chart unavailable (SR0110.3.1.1)** user-triggered exception (page 103) needs to be recorded before the phase can be completed. ■ Otherwise, phase is completed.

Resume phase (SR0110.2.4)

- Function: Resuming of phase
  The phase was in the **Active mode (SR0110.1.2)** status (page 94) when the Production Execution Client was shut down.
- Trigger: At restart of the Production Execution Client, phase is resumed.
- Postcondition: Phase is active

Step	#	Description
System resumes phase	10	Phase displays the chart image placeholder with the embedded text **No chart generated**.
Display saved chart	20	If the chart has been generated and stored before, phase loads and displays the saved chart image.

Process Parameters (SR0110.8+)

The following process parameters define the behavior of the phase.

BASIC PARAMETERS

Instruction (SR0110.8.1)

➢ For recent changes, see revision history (page 115).

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	

Identified equipment entity (SR0110.8.2)

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

CHART PARAMETERS

Chart plot (SR0110.8.3)

The attributes are available for each of the 16 **Chart plot** process parameters (Plot 1 - Plot 16).

Attribute	Type	Comment
Enabled	Flag	Controls if the plot configuration is used to draw a plot.
Property	String	Historian point to be read.
Query template	Choice list	Defines the template to be used for data retrieval. By default, the system provides three templates: **Raw archive data** provides all archived data for the history period. **Plot data** provides the data dedicated to plotting (trending) applications within the history period. **Marker (DigitalState)** provides the string values within the history period that are available for use as segment markers.
Plot format	Choice list	Defines the plot drawing to be used with regard to line color, thickness, and line pattern.
Timestamp 1	Timestamp	Optional parameter to be passed to the query template for query-specific usage. Evaluated as data retrieval start date and time for system-defined query templates (**Raw archive data**, **Plot data**, **Marker (DigitalState)**).

Attribute	Type	Comment
Timestamp 2	Timestamp	Optional parameter to be passed to the query template for query-specific usage. Evaluated as data retrieval end date and time for system-defined query templates (**Raw archive data**, **Plot data**, **Marker (DigitalState)**).
String 1	String	Optional parameter to be passed to the query template for query-specific usage.
String 2	String	Optional parameter to be passed to the query template for query-specific usage.
Duration	Duration	Optional parameter to be passed to the query template for query-specific usage.
Long	Long	Optional parameter to be passed to the query template for query-specific usage.
MeasuredValue 1	MeasuredValue	Optional parameter to be passed to the query template for query-specific usage.
MeasuredValue 2	MeasuredValue	Optional parameter to be passed to the query template for query-specific usage.

Property Selection editor (Framework capability)

The system provides a Property Selection editor for selecting an equipment property based on its data type (numeric, string, boolean).

Chart axis (SR0110.8.4)

The attributes are available for each of the 4 **Chart axis** process parameters (Axis 1 - Axis 4).

Attribute	Type	Comment
Enabled	Flag	Controls if the axis configuration is used to draw an axis.
Unit of measure	Choice list	Defines the unit of measure for the axis and thus its label.

Attribute	Type	Comment
Minimum	Numeric	The minimum value shown on the y-axis. A value is required if **Autorange** is disabled.
Maximum	Numeric	The maximum value shown on the y-axis. A value is required if **Autorange** is disabled.
Autorange	Flag	Defines if the minimum and maximum values of the y-axis are to be determined automatically. If so, the values defined for the **Minimum** and **Maximum** attributes are ignored. Default setting: **Yes**
Scaling mode	Choice list	Defines the scaling mode. Available settings: **Linear**, **Logarithmic**. Default setting: **Linear**.
Number format	String	Defines the format pattern for numeric values. Examples: **0**: Integer portion of a number. **0.0**: Integer portion and one fractional digit. **0.00**: Integer portion and two fractional digits. Default setting: **0**.

CONFIGURATION OF USER-TRIGGERED EXCEPTIONS

Chart unavailable (SR0110.8.5)

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**.

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 **Chart unavailable (SR0110.3.1.1)** user-triggered exception (page 103).

Exceptions (SR0110.3+)

The phase supports user-defined, user-triggered (page 103), system-triggered (page 103), and post-completion exceptions (page 104) and their configuration by means of process parameters (page 99).

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 (SR0110.3.1+)

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.

Chart unavailable (SR0110.3.1.1)

The **Chart unavailable** exception allows an operator to confirm the phase without a chart.

Representation during exception handling:

- Instruction:
  Confirm without chart.
  Confirm button.
- Exception text:
  <Exception text>
  (taken from **Chart unavailable (SR0110.8.5)** process parameter (page 102))
  - Example:
    Confirmed without chart.

Chart unavailable - Logic (SR0110.3.1.1.1)

- Trigger: Exception is selected
- Postcondition: Phase can be confirmed

Step	#	Description
Operator confirms exception	10	Phase shows exception description to be signed according to **Chart unavailable (SR0110.8.5)** process parameter (page 102).
Operator signs exception	20	Phase records exception.
Phase activation	30	Phase returns to the **Active mode (SR0110.1.2)** layout (page 94) and disables the **Reload** button.

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 (SR0110.3.6+)

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.

Invalid configuration error (SR0110.3.6.1)

UI text	Comment
One or more configuration errors have occurred:	Message pack: PhaseShwHstDatChrt<version> Message ID: InvalidConfigurationError_Category

UI text	Comment
Entity not defined.	Message pack: PhaseShwHstDatChrt<version> Message ID: EntityUndefined_ErrorMsg
<entity identifier> entity has no <property identifier> property. (<parameter ID>) ... <entity identifier> entity has no <property identifier> property. (<parameter ID>)	Message pack: PhaseShwHstDatChrt<version> Message ID: EntityHasNoProperty_ErrorMsg
No Y-axis for <UoM> defined for <property identifier> property. (<parameter ID>) ... No Y-axis for <UoM> defined for <property identifier> property. (<parameter ID>)	Message pack: PhaseShwHstDatChrt<version> Message ID: UOMUndefinedForYAxis_ErrorMsg
No Y-axis without unit of measure defined for <property identifier> property. (<parameter ID>) ... No Y-axis without unit of measure defined for <property identifier> property. (<parameter ID>)	Message pack: PhaseShwHstDatChrt<version> Message ID: NoUOMUndefinedForYAxis_ErrorMsg
Too many different units of measure defined to be displayed on available Y-axes.	Message pack: PhaseShwHstDatChrt<version> Message ID: TooManyUOMsDefined_ErrorMsg
A unit of measure has been defined more than once for the Y-axes.	Message pack: PhaseShwHstDatChrt<version> Message ID: DuplicatedUOMForYAxis_ErrorMsg
Query template not defined. (<parameter ID>, ...)	Message pack: PhaseShwHstDatChrt<version> Message ID: UndefinedQueryTemplate_ErrorMsg
Property not defined. (<parameter ID>, ...)	Message pack: PhaseShwHstDatChrt<version> Message ID: UndefinedPropertyType_ErrorMsg
Plot format not defined. (<parameter ID>, ...)	Message pack: PhaseShwHstDatChrt<version> Message ID: UndefinedPlotFormat_ErrorMsg
Range not defined. (<parameter ID>, …)	Message pack: PhaseShwHstDatChrt<version> Message ID: UndefinedRange_ErrorMsg

Data retrieval error (SR0110.3.6.2)

UI text	Comment
Cannot retrieve the requested data. Please contact your system administrator.	Message pack: PhaseShwHstDatChrt<version> Message ID: ChartGeneration_ErrorMsg

Chart unavailable error (SR0110.3.6.3)

UI text	Comment
Cannot complete the phase, since there is no chart available.	Message pack: PhaseShwHstDatChrt<version> Message ID: ChartUnavailable_ErrorMsg

Output Variables (SR0110.9+)

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.

Chart available (SR0110.9.1)

- Data type: Boolean
- Usage: The output variable states if the chart is available (`TRUE`) or not (`FALSE`).

Configuration Keys (SR0110.11+)

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

Chart resolution (SR0110.11.1)

- **Phase/ShowHistoricalDataChartPhase/HistoricalDataChartReportResolution**
- **Type**: String
- **Value**: 300
- **Description**: Defines the chart resolution in DPI for batch report printout.
- **Range**: N/A

Anti-aliasing for plots (SR0110.11.4)

- **Phase/ShowHistoricalDataChartPhase/HistoricalDataChartsetPlotAntiAlias**
- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, anti-aliasing is enabled for plots.
- **Range**: N/A

Anti-aliasing for text (SR0110.11.5)

- **Phase/ShowHistoricalDataChartPhase/HistoricalDataChartsetTextAntiAlias**
- **Type**: Boolean
- **Value**: False
- **Description**: If the value is set to **true**, anti-aliasing is enabled for text.
- **Range**: N/A

Plot renderers (SR0110.11.2)

- **Phase/ShowHistoricalDataChartPhase/PlotRenderers**
- **Type**: Object - List
- **Value**:

```
<?xml version="1.0" encoding="UTF-8"?>
<plotRenderers xmlns="http://www.rockwell.com/mes/commons/base/graph">
<plotRenderer name="Black" description="Black solid 1.5pt">
  <paint red="0" green="0" blue="0"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Red" description="Red solid 1.5pt">
  <paint red="255" green="0" blue="0"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Green" description="Greed solid 1.5pt">
  <paint red="32" green="172" blue="32"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Blue" description="Blue solid 1.5pt">
  <paint red="0" green="0" blue="255"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Magenta" description="Magenta solid 1.5pt">
  <paint red="255" green="0" blue="255"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Cyan" description="Cyan solid 1.5pt">
  <paint red="0" green="223" blue="218"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Orange" description="Orange solid 1.5pt">
  <paint red="225" green="170" blue="40"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Light green" description="Light green solid 1.5pt">
  <paint red="0" green="226" blue="0"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Light blue" description="Light blue solid 1.5pt">
  <paint red="33" green="160" blue="223"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Purple" description="Purple solid 1.5pt">
  <paint red="161" green="67" blue="255"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Light red" description="Light red solid 1.5pt">
  <paint red="255" green="125" blue="125"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Petrol" description="Petrol solid 1.5pt">
  <paint red="18" green="157" blue="140"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Brown" description="Brown solid 1.5pt">
  <paint red="124" green="90" blue="76"/>
  <stroke/>
</plotRenderer>
```

```
<plotRenderer name="Tan" description="Tan solid 1.5pt">
  <paint red="179" green="175" blue="13"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Livid" description="Livid solid 1.5pt">
  <paint red="113" green="135" blue="199"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Dark red" description="Dark red solid 1.5pt">
  <paint red="192" green="80" blue="77"/>
  <stroke/>
</plotRenderer>
<plotRenderer name="Tan dash dot" description="Tan dash-dotted 1.5pt">
  <paint red="179" green="175" blue="13"/>
  <stroke lineJoin="1" miterLimit="3.0">
    <dashArray>6.0</dashArray>
    <dashArray>3.0</dashArray>
    <dashArray>1.0</dashArray>
    <dashArray>3.0</dashArray>
  </stroke>
</plotRenderer>
<plotRenderer name="Black dashed" description="Black dashed 1.5">
  <paint red="0" green="0" blue="0"/>
  <stroke>
    <dashArray>6.0</dashArray>
    <dashArray>3.0</dashArray>
  </stroke>
</plotRenderer>
</plotRenderers>
```

- **Description**: Specifies the list of plot renderers available for the plot-related process parameters of the **Show historical data chart** phase.
  For more information, please refer to section "Configuring Plot Styles for Historical Data Charts", chapter "Administration" in "Technical Manual Phases of the Equipment Automation Package" [A4] (page 113).
- **Range**: N/A

Query templates (SR0110.11.3)

- **Phase/ShowHistoricalDataChartPhase/QueryTemplates**
- **Type**: Object - List
- **Value**:

```
<QueryTemplates xmlns="http://www.rockwell.com/mes/commons/base/query">
  <QueryTemplate name="Raw archive data" description="All raw archive values within defined
          period." usage="chart">
    <Parameters>
      <Parameter name="Property" description="pi-point" datatype="String"
          systemdefined="true"/>
      <Parameter name="Timestamp1" description="start-time" datatype="DateTime"/>
      <Parameter name="Timestamp2" description="end-time" datatype="DateTime"/>
    </Parameters>
    <Outputs>
      <Parameter name="value" description="numeric tag value" datatype="Float"
          systemdefined="true"/>
      <Parameter name="time" description="time stamp" datatype="DateTime"
```

```
          systemdefined="true"/>
    </Outputs>
    <Query provider="OSI_PI">SELECT if status = 0 THEN value ELSE null "value", time FROM
          piarchive..picomp WHERE tag=%Property% AND time BETWEEN %Timestamp1% AND
          %Timestamp2%</Query>
  </QueryTemplate>
  <QueryTemplate name="Plot data" description="Data dedicated to plotting (trending)
          applications within defined period." usage="chart">
    <Parameters>
      <Parameter name="Property" description="pi-point" datatype="String"
          systemdefined="true"/>
      <Parameter name="Timestamp1" description="start-time" datatype="DateTime"/>
      <Parameter name="Timestamp2" description="end-time" datatype="DateTime"/>
      <Parameter name="intervalCount" description="#ofPixels" datatype="Long"
          systemdefined="true"/>
    </Parameters>
    <Outputs>
      <Parameter name="value" description="numeric tag value" datatype="Float"
          systemdefined="true"/>
      <Parameter name="time" description="time stamp" datatype="DateTime"
          systemdefined="true"/>
    </Outputs>
    <Query provider="OSI_PI">SELECT if status = 0 THEN value ELSE null "value", time FROM
          piarchive..piplot WHERE tag=%Property% AND time BETWEEN %Timestamp1% AND
          %Timestamp2% AND intervalCount = %intervalCount%</Query>
  </QueryTemplate>
  <QueryTemplate name="Marker (DigitalState)" description="All digital state string values
          for chart within defined period used as segment markers." usage="chart">
    <Parameters>
      <Parameter name="Property" description="pi-point" datatype="String"
          systemdefined="true"/>
      <Parameter name="Timestamp1" description="start-time" datatype="DateTime"/>
      <Parameter name="Timestamp2" description="end-time" datatype="DateTime"/>
    </Parameters>
    <Outputs>
      <Parameter name="value" description="string tag value" datatype="String"
          systemdefined="true"/>
      <Parameter name="time" description="time stamp" datatype="DateTime"
          systemdefined="true"/>
    </Outputs>
    <Query provider="OSI_PI">SELECT DIGSTRING(status) "value", time FROM piarchive..picomp
          WHERE tag=%Property% AND time BETWEEN %Timestamp1% AND %Timestamp2%</Query>
  </QueryTemplate>
</QueryTemplates>
```

- **Description**: Specifies the list of query templates available for the plot-related process parameters of the **Show historical data chart** phase.
  For more information, please refer to section "Configuring Query Templates for Historical Data Charts", chapter "Administration" in "Technical Manual Phases of the Equipment Automation Package" [A4] (page 113).
- **Range**: N/A

Provide shortcuts (SR0110.11.6)

- **Phase/ShowHistoricalDataChartPhase/ProviderShortcuts**
- **Type**: List
- **Value**: (PiConnector=OSI_PI, PiMockConnector=CSV_MOCK)
- **Description**: Defines shortcuts for providers. The shortcuts can be used within QueryTemplateXML, e.g. <Query provider="OSI_PI">.
  Each entry must be formatted as follows: key=value, e.g. PiConnector=OSI_PI
- **Range**: N/A

Performance (SR0110.12+)

Reference scenario

The reference scenario holds a total of:

- 16 Historian properties

  Data volume perspective:

  - 4 properties with less than 100 data points retrieved
  - 5 properties with more than 2,000 data points retrieved
  - 3 properties with more than 4,000 data points retrieved
  - 4 properties with more than 20,000 data points retrieved

  Query template perspective:

  - 8 properties using the **Raw archive data** query template
  - 6 properties using the **Plot data** query template
  - 2 properties using the **Marker (DigitalState)** query template
- 3 different query templates for data retrieval (**Raw archive data**, **Plot data**, **Marker (DigitalState)**)

  Example for **Raw archive data** query template:

  - `SELECT if status = 0 THEN value ELSE null "value", time FROM piarchive..picomp WHERE tag=%Property% AND time BETWEEN %Timestamp1% AND %Timestamp2%Each: 3 subsequent operations`
- 14 days as duration for data retrieval

The setup of the Historian Infrastructure for PharmaSuite is performed using the normal access topology as defined in Automation Integration Configuration Scenarios (see "Technical Manual Installation - Enterprise Edition" [A3] (page 113)).

Performance of Chart Rendering (SR0110.12.1)

Based on the Reference scenario (page 111), the rendering of a trend chart does not take longer than 10 seconds in the PharmaSuite system test environment. The duration is measured between activation of the phase and displaying the chart image.

Reference Documents

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

No.	Document Title	Part Number
A1	PharmaSuite Functional Requirement Specification Execution Framework	PSFRSEF-RM004E-EN-E
A2	PharmaSuite Functional Requirement Specification Recipe and Workflow Management	PSFRSRD-RM008E-EN-E
A3	PharmaSuite Technical Manual Installation - Enterprise Edition	PSEN-IN008E-EN-E
A4	PharmaSuite Technical Manual Phases of the Equipment Automation Package	PSEA-PM004B-EN-E

TIP

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

Appendix A

Document Information

The document information covers various data related to the document.

Approval

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

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

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

Version Information

Object	Version
PharmaSuite	8.4
Get OPC values	1.0 MR3
Set OPC values	1.0 MR3
Monitor numeric value	1.0 MR4
Get alarms	1.0 MR4
Show historical data chart	1.0 MR5
Functional Requirement Specification	1.0

Revision History

The following table describes the history of this document.

Changes related to the document:

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

Changes related to "Get OPC Values Phase" (page 3):

Object	Description	Document
System-triggered Exceptions (SR0341.3.2+) (page 19)	Update The message dialog of a system-triggered exception no longer provides a **Cancel** button.	1.0
Override Recorded Value - Logic (SR0341.3.1.1.1) (page 21)	Update Step 30.1: The message dialog no longer provides a **Cancel** button. Step 30.1.2: Phase requires to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).	1.0
Override Recorded Value - Logic (SR0341.3.1.2.1) (page 23)	Update Step 30.1: The message dialog no longer provides a **Cancel** button. Step 30.1.2: Phase requires to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).	1.0
Override Recorded Value - Logic (SR0341.3.1.3.1) (page 24)	Update Step 30.1: The message dialog no longer provides a **Cancel** button. Step 30.1.2: Phase requires to sign an **Exception canceled** exception and then allows the operator to return to the user-triggered exception view (Step 20).	1.0
Instruction (SR0341.8.1) (page 12)	Update The maximum length of the **Instruction** process parameter is 2000 characters (including HTML tags). No change of code.	1.0

Changes related to "Set OPC Values Phase" (page 35):

Object	Description	Document
Instruction (SR0342.8.1) (page 43)	Update The maximum length of the **Instruction** process parameter is 2000 characters (including HTML tags). No change of code.	1.0

Changes related to "Monitor Numeric Value Phase" (page 63):

Object	Description	Document
System-triggered Exceptions (SR0360.3.2+) (page 72)	Update The message dialog of a system-triggered exception no longer provides a **Cancel** button.	1.0
Instruction (SR0360.8.1) (page 68)	Update The maximum length of the **Instruction** process parameter is 2000 characters (including HTML tags). No change of code.	1.0

Changes related to "Get Alarms Phase" (page 79):

Object	Description	Document
System-triggered Exceptions (SR0365.3.2+) (page 87)	Update The message dialog of a system-triggered exception no longer provides a **Cancel** button.	1.0
Instruction (SR0365.8.1) (page 84)	Update The maximum length of the **Instruction** process parameter is 2000 characters (including HTML tags). No change of code.	1.0

Changes related to "Show Historical Data Chart" (page 93):

Object	Description	Document
Instruction (SR0110.8.1) (page 99)	Update The maximum length of the **Instruction** process parameter is 2000 characters (including HTML tags). No change of code.	1.0

Index

G

M

S