LISTEN.
THINK.
SOLVE.®

PharmaSuite®

EXECUTION FRAMEWORK

RELEASE 8.4

FUNCTIONAL REQUIREMENT SPECIFICATION

PUBLICATION PSFRSEF-RM004E-EN-E-DECEMBER-2017
Supersedes publication PSFDEF-RM004D-EN-E

Allen-Bradley • Rockwell Software

Rockwell Automation

Contact Rockwell See contact information provided in your maintenance contract.

Copyright Notice

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

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

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

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

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

ControlNet is a registered trademark of ControlNet International.

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

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

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

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

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

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

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

Contents

Figures

Chapter 1

Introduction

This document details the requirements of functions that are available through the functionality of PharmaSuite phase building blocks as well as through the Production Execution Client. However, the functions are implemented by the execution framework. The framework supports the electronic execution and recording of recipes and workflows, monitoring and recovery capabilities, and the exception management.

In addition, the Phase-related Common Capabilities (page 81) chapter describes the common requirements of all PharmaSuite phase building blocks.

Each requirement is composed of a name and a unique identifier (e.g. Phase column (SR0900.4.1)). If a requirement's meaning is for requirement grouping only, the identifier is appended by a plus sign (e.g. Representation in Navigator (SR0900.4+)).

In some cases, additional context information is available, indicated in the document by a frame and a gray background color. This context information is related to the respective requirement, but not part of the formal requirement description.

The revision history (page 89) 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).

Production Execution Client (SR3071.4+)

The Production Execution Client supports the processing of recipes (designed with Recipe Designer), workflows (designed with Workflow Designer), and pre-defined workflows on the shop floor.

For details regarding pre-defined workflows, please refer to the **GUI components and navigation for pre-defined workflows (SR3071.4.2+)** requirement (page 18) and the "Functional Requirement Specification Production Execution" [A3] (page 87).

Task-oriented UI (SR3071.4.3)

Based on information about its work center, station, device, and the logged-in user, the Production Execution Client displays a list of startable or restartable tasks in the **Cockpit (SR3071.4.1.1)** view (page 6).

Processing Types (SR3071.4.3.2)

The Production Execution Client supports:

- Batch processing (based on a master recipe designed with Recipe Designer - Batch, [A2] (page 87))

  Related tasks in the **Production** group of the **Cockpit (SR3071.4.1.1)** view are **Batch Processing** and **Material Processing**. Both open a specific Selection Window to start processing.

- Device processing (based on a master recipe designed with Recipe Designer - Device, [A2] (page 87))

  Related task in the **Production** group of the **Cockpit (SR3071.4.1.1)** view is **Device Processing**. It opens a Selection Window to start processing.

- Workflow processing (based on a master workflow designed with Workflow Designer, [A2] (page 87))

  Related task in the **Production** group of the **Cockpit (SR3071.4.1.1)** view is **Workflow Processing**. It opens a Selection Window to start processing.

- One-click workflows assigned to a particular work center (based on a master workflow designed with Workflow Designer, [A2] (page 87))

  Related tasks in the **Cockpit (SR3071.4.1.1)** view are defined in Workflow Designer.

- Pre-defined workflows assigned to a particular work center (e.g. material receipt)

  Related tasks are available in the **Inventory** group of the **Cockpit (SR3071.4.1.1)** view.

Framework (SR3071.4.1)

PharmaSuite provides the following execution-specific framework capabilities for recipes and workflows designed with Recipe and Workflow Designer.

GUI Areas (SR3071.4.2.4)

The UI of the Production Execution Client consists of three areas:

- **header bar (SR3071.4.2.4.1)** area (page 4),
- work area of the **Execution Window (SR3071.4.2.4.2)** view (page 12), the **Cockpit (SR3071.4.1.1)** view (page 6), the **Navigator (SR3071.4.2.3)** view (page 9), and
- **control bar (SR3071.4.2.4.3)** area (page 6).

Header Bar (SR3071.4.2.4.1)

The system provides a header bar that displays execution-specific information as listed below:

- In the **Cockpit (SR3071.4.1.1)** view (page 6) and the **Navigator (SR3071.4.2.3)** view (page 9):
  - logged-in user
  - current work center
  - current station
- **Batch and device** processes, in the **Execution Window (SR3071.4.2.4.2)** view (page 12):
  - order step identifier (order, unit procedure), unit procedure instance count ((if > 1)
  - batch identifier
  - serial number (for device-specific orders only)
  - operation identifier

  Operations of looped processes can be run and completed several times. In these cases the system appends a count to their operation identifiers that indicates how often they have been run.

- **Workflows,** in the **Execution Window (SR3071.4.2.4.2)** view (page 12):
  - workflow step identifier (workflow, unit procedure)
  - [workflow] Processing name
  - operation identifier

  Operations of looped processes can be run and completed several times. In these cases the system appends a count to their operation identifiers that indicates how often they have been run.

- In the **exception management (SR1079+)** view for phases and operations according to the **List of Exceptions (SR1079.2.2)** feature (page 75):
  - order step identifier (order, unit procedure), unit procedure instance count (if > 1), batch identifier (for batch- and device-specific orders only), serial number (for device-specific orders only)
    or
    workflow step identifier (workflow, unit procedure), unit procedure instance count (if > 1), [workflow] processing name
  - operation identifier
  - phase identifier (only for exceptions recorded related to a phase)
- In the **exception management (SR1079+)** view for unit procedures according to the **Unit procedure-specific Exceptions (SR1079.2.2.1)** feature (page 76):
  - order step identifier (order, unit procedure), batch identifier (for batch- and device-specific orders only), serial number (for device-specific orders only)
    or
    workflow step identifier (workflow, unit procedure), [workflow] processing name
  - unit procedure identifier and instance count (if > 1)
- For all listed views, the system supports notification bars specific to
  - escalation management, see **Graphical Representation of Escalation States (SR1089.6.3.2)** details (page 39) and
  - paused unit procedures, see **Graphical Representation of a Paused Unit Procedure (SR1089.8.3.1)** details (page 33).

In a Selection Window, the header bar displays a selection-specific title and the identifiers of affected objects, if applicable.

Control Bar (SR3071.4.2.4.3)

The system provides a control bar that enables the navigation between different views and within a view as listed below:

- **Cockpit (SR3071.4.1.1)** view (page 6)
- **Navigator (SR3071.4.2.3)** view (page 9)
- **Exception management (SR1079+)** view (page 67)
- Page up
- Page down
- Switch between parallel running operations

The navigation is supported by keyboard shortcuts.

Cockpit (SR3071.4.1.1)

➢ For recent changes, see revision history (page 89).

The system allows to define an access privilege for the protection of master recipes and master workflows from unauthorized access. Subsequently, it shall only allow to select an order or workflow whose access privilege matches the access privilege of the logged-in user.

The Cockpit represents an operator's entry point to the execution process. Specific to each station and user, the Cockpit provides a list of all startable, resumable, and running processes, from which the operator can select unit procedures and operations to start or switch to. The running processes also include those operations of the same unit procedure that are running at another station. For an idle operation, its creation time is displayed close to the operation identifier.

In an exceptional situation it may be necessary to repair a corrupted process (order, workflow) by unloading it from the EBR server. The Cockpit also lists the unit procedures and operations of an unloaded process even though they cannot be processed. For details related to unloaded processes, see **Unload and Reload Procedures of Orders and Workflows (SR1200.2.2)** feature (page 58).

Definition of unit procedure statuses (and equivalent order step statuses):

- **Idle** (**Generated**): The unit procedure (order step) can be selected for processing.
- **Running** (**In process**): The unit procedure is visible in the Cockpit. Its operations are idle or running.
- **Held** (**Held**): A running unit procedure has been detached from its work center. It can be resumed at any allowed work center.
- **Paused** (order step keeps its current status): A running unit procedure has been paused, for example to hold triggering activities. It can only be resumed at its

work center.

- **Aborted** (**Aborted**): A running unit procedure that has been aborted. It can be reactivated to allow further processing at any allowed work center.
- **Completed** (**Finished**): The processing of the last operation has been completed.

Definition of operation statuses:

- **Idle**: The operation is visible in the Cockpit and ready for starting its phases.
- **Running**: At least one phase of the operation is running.
- **On hold**: At least one phase of the operation has been started, but the current phase is on hold. For example, detaching an operation puts it on hold.
- **Resumed**: Same as a running operation, but the operation has been detached and resumed before.

The process and operation types (startable, not startable, running/resumable) are indicated by colors.
For running operations, the name of the computer where the Production Execution Client is running, is displayed close to the operation identifier.

For operation-related capabilities, see:

- **Detaching an operation (SR1089.6.1+)** details (page 35)
- **Triggering an operation by events (SR1089.6.2+)** details (page 36)
- **Escalation management of event-triggered operations (SR1089.6.3+)** details (page 38)
- **Server-side running operations (SR1089.6.4+)** details (page 39)
- **Auto-startable operations (SR1089.6.5)** details (page 40)

For unit procedure-related capabilities, see:

- **Detaching a unit procedure (SR1089.8.4+)** details (page 30)
- **Context-related action (SR1089.8.1)** details (page 31)
- **Appending workflows to a unit procedure (SR1089.8.2)** details (page 31)
- **Pausing a unit procedure (SR1089.8.3)** details (page 33)
- **Reactivating a unit procedure (SR1089.8.5)** details (page 34)

Running or Resumable Processes

Information of running unit procedures (SR3071.4.1.1.6)

The system provides the following information specific to running unit procedures.

Master recipe-related unit procedures for batches:

- Batch identifier
- Order step identifier (order, unit procedure)

- Unit procedure instance count (only if > 1)

Master recipe-related unit procedures for devices:

- Serial number
- Order step identifier (order, unit procedure)
- Unit procedure instance count (only if > 1)

Workflow-related unit procedure:

- [Workflow] Processing name
- Workflow step identifier (workflow, unit procedure)

Indicator for running operations (SR3071.4.1.1.5)

The system highlights running operations that have already been started and that currently are not detached.

Parallel operations (SR3071.4.1.1.1)

If more than one operation has been started in parallel, the operator can switch between the running operations (see **Control Bar (SR3071.4.2.4.3)** area (page 6)).

Sequence of idle operations (SR3071.4.1.1.2)

Idle operations are presented in a defined sequence according to the SFC model of the recipe or workflow. In parallel branches, the presentation of the idle operations starts with the leftmost branch.

Scale communication-related locking (SR3071.4.1.1.3)

If a scale communication is running within the currently active process, the system does not allow to start another or to switch to any other process.

Startable Processes

Grouping (SR3071.4.1.1.4)

One-click workflows are grouped according to the defined type in the Workflow Designer.
See also **Workflow - Header Attributes (SR3146.9.12.1)** in "Functional Requirement Specification Recipe and Workflow Management" [A2] (page 87).

Navigator (SR3071.4.2.3)

➢ For recent changes, see revision history (page 89).

The Navigator represents an operator's information and service point during the execution process. Specific to each unit procedure, the Navigator provides a historical view of the executed process steps. For each step, the operator can access basic and detail information as well as additional actions.

- The Phase column presents
  - a history view of the completed phases of the given unit procedure and
  - the active steps of the current operation.

  > The statuses (active, completed) are indicated by colors.
  > The identifier of the operation to which a phase belongs is displayed along with the phase identifier.
  > Phases and operations of looped processes can be run and completed several times. In these cases the system appends a count to their identifiers that indicates how often they have been run.

- The Information column
  - displays some key information for each completed phase (e.g. batch and sublot that was identified) and
  - provides access to the **Detail Information (SR3071.4.2.3.3)** window (page 10) per phase (if applicable).

  > Phases for which an exception has been recorded are displayed with a red upper left corner as exception marker.

- The Action column either
  - provides access to phase-specific actions that can be performed related to completed phases (if applicable, e.g. reprint of a sublot label as a **post-completion (SR1079.1.3)** exception (page 70)) or
  - provides access to general recovery actions that can be performed related to active phases (**recovery (SR1200.3+)** capabilities (page 57)).

Operations Running on Different Devices (SR3071.4.2.3.6)

For idle operations of the current unit procedure and the current device, the access to the phases of the operation is disabled in the Navigator. This applies to all columns, especially to post-completion exceptions.

Loop Counter (SR3071.4.2.3.2)

Operations of looped processes can be run and completed several times. In these cases the system appends a count to their operation identifiers that indicates how often they have been run.

- First execution: no counter
- Second execution: counter = 2
- etc.

Detail Information (SR3071.4.2.3.3)

Through the Navigator, the system provides detail information per phase. The information includes:

- For batch and device processes:
  - batch record-related header data (incl. batch, order, and product information) or
    device record-related header data (incl. serial number, batch, order, and product information).
- For workflows:
  - workflow record-related header data (incl. processing name, workflow).
- Data of the phase, as presented within the specific report
- List of recorded exceptions (see **Exception management (SR1079+)** view (page 67))
- Input panel for new exceptions or exception-related comments (see **Exception management (SR1079+)** view (page 67))

Data Panel for Batch- and Device-related Processes (SR3071.4.2.3.4)

The data panel of the Navigator displays data of the:

- order identifier,
- batch identifier,
- unit procedure identifier and instance count (if > 1),
- serial number (for device-specific orders only),
- material identifier, and
- material short description.

Data Panel for Workflows (SR3071.4.2.3.5)

The data panel of the Navigator displays data of the:

- workflow identifier,

- [workflow] Processing name, and
- unit procedure identifier and instance count (if > 1).

Cancel a Workflow (SR3071.4.2.3.7)

The data panel for workflows provides a control to cancel a workflow.

- If the workflow is linked to a unit procedure of a batch- or device-specific order, the workflow cannot be canceled unless the link has been removed. The system displays an appropriate message.
- When a workflow is canceled,
  - the system sets the status of a workflow to **Canceled** (see **Status Management of Workflows (SR1085.3)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87));
  - the status transition is propagated to its workflow steps in the **In process** status (see **Cancel Workflow - Status Propagation (SR1085.2.1.2)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87));
  - the system ensures that its workflow steps cannot be started and/or executed anymore (see **Cancel Workflow - No Workflow Step Execution (SR1085.2.1.3)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87));
  - all related phases, operations, unit procedures, and procedures are terminated automatically and are no longer visible as running processes in the Cockpit of the Production Execution Client.
    At each of the related Production Execution Clients, the system displays an appropriate message.
    Termination of related phases and operations also applies to server-run operations, which run on the Operation Execution server without any user interaction.
    See **Cancel Workflow - Termination of Procedural Elements (SR1085.2.1.3.1)** "Functional Requirement Specification Runtime Data Management" [A4] (page 87).
  - The system automatically records a "cancel workflow"-specific exception for all active phases of the workflow. The risk level of the exception can be configured.

For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).
For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Execution Window (SR3071.4.2.4.2)

The Execution Window represents an operator's workspace during the execution process. Here, the operator performs the execution-related actions and registers process-relevant data.

Buttons can be placed in the work area in order to confirm the completion of the current step.

> Phases for which an exception has been recorded are displayed with a red upper left corner as exception marker.

Listed Phases (SR3071.4.2.4.2.1)

The work area displays active phases for execution, as well as recently completed previous phases (of the same operation) and subsequent phases that are not active yet. Subsequent phases are shown up to the next branching that represents a decision point.

Active Phases (SR3071.4.2.4.2.2)

The work area displays the phases according to their status:

- Preview of a phase: disabled

  > If the required information or input is not available yet, the preview shows a corresponding system message.

- Active phase(s): highlighted
- Completed phase(s): disabled with OK symbol

Parallel Phases (SR3071.4.2.4.2.3)

During execution, the work area displays parallel phases in a defined sequence, starting with the left phase of the parallel branch.

Preview of Phases (SR3071.4.2.4.2.4)

During execution, the preview of phases behaves as listed below:

- For parallel branches, the sequence of preview steps is not changed (no re-ordering).
- For loops, the active loop phase appears just after the previous phase that was executed before the loop started.

Scrolling (SR3071.4.2.4.3.1)

The Page up and Page down capabilities apply to any kind of information provided by phases, so that information is available to the operator regardless of the size of a phase UI.

Actions of the System Group (SR3071.4.8)

The **System** group of the Cockpit provides the following actions:

- Change User
- Register at Station
- About
- Exit

- The **Help** action provides access to the help system.

Register at Station Constraints (SR3071.4.8.1)

The **Register at Station** action is only allowed if all running operations are either completed or put on hold (e.g. detached) and no other process or workflow is running.

Register at Station Configuration and Scanner Support (SR3071.4.8.2)

The **Register at Station** action can be triggered in the following ways:

- operator taps the button or
- operator scans a station barcode.

For both ways, it can be configured separately which user has access to the action and on which station the action is available.

User Access to Station Constraints (SR3071.4.8.3)

➢ For recent changes, see revision history (page 89).

To run the Production Execution Client on a specific station, a user must be authorized according to the access privilege assigned to the station. This applies to:

- Login
- User change
- Register at station by dialog
- Register at station by barcode scan.

The system allows to define an access privilege for the protection of master recipes and master workflows from unauthorized access. Subsequently, it shall only allow to run the Production Execution Client on a specific station if the currently running processes and workflows are not protected by an access privilege or their access privilege matches the access privilege of the logged-in user.

USER CHANGE CONSTRAINTS (SR3071.4.8.3.1)

The following business logic applies to the **Change User** action if the user who logs in is not authorized to run the Production Execution Client on the current station:

- If all previously running operations have been set on hold (e.g. detached) and no other processes or workflows are running, the system allows to log in, but only the actions of the **System** group are available in the Cockpit.
- If there are still operations, processes, or workflows running, the system does not allow the user to log in.

EXIT CONSTRAINTS (SR3071.4.8.5)

The **Exit** action is only allowed if all previously running operations have been set on hold (e.g. detached) and no other process or workflow is running.

Identify Resources by Label Information (SR1094.17+)

The system supports the identification of resources by their label information.

Usage of identification devices (SR1094.17.1+)

The system supports the usage of barcode reading devices in order to identify resources.

Barcode symbologies (SR1094.17.1.1)

The system validates and processes the following barcode symbologies, at minimum by supporting the devices that can process these barcode encoding standards:

- Code 128
- Code 39

Incorrect barcode (SR1094.17.1.1.1)

➢ Applies to pre-defined inventory workflows only.

The system does not allow the processing of wrongly coded barcode information.

Unique sublot (SR1094.17.1.1.2+)

Unique sublot/barcode

Single barcode - Unique Sublot (SR1094.17.1.1.2.1)

➢ Applies to pre-defined inventory workflows only.

The system ensures that a label scan performed by the manufacturing operator only leads to the identification of one unique sublot.

Update data after identification (SR1094.25)

➢ Applies to pre-defined inventory workflows only.

After a successful identification of an input material, the system stores the data related to the identification step.

> For details of material identification with barcode scanning during order execution, see **Identify material phase (SR0200)** in "Functional Requirement Specification Dispense and Inline Weighing" [A8] (page 87) and **Identify material phase (SR0050)** in "Functional Requirement Specification Material Tracking Phases" [A9] (page 87).

Start Processing by Scanning (SR3071.4.9+)

Start Batch Processing (SR3071.4.9.1)

In the **Cockpit (SR3071.4.1.1)** view (page 6) and the Batch Processing Selection Window view, the system allows an operator to start the processing of a unit procedure by scanning a batch-related barcode. The barcode corresponds to the batch identifier of the material to be produced.

The system displays an appropriate message in case a unit procedure cannot be started due to the following situations:

- No unit procedure for the scanned batch is available for processing at the current work center.
- For the scanned batch, more than one unit procedure is available for processing at the current work center. This applies to parallel unit procedures on the same work center or to non-unique target batches for different orders.

Start Device Processing (SR3071.4.9.2)

In the **Cockpit (SR3071.4.1.1)** view (page 6) and the Device Processing Selection Window, the system allows an operator to start the processing of a unit procedure by scanning a serial number-related barcode. The barcode corresponds to a specific device to be produced.

The system displays an appropriate message in case a unit procedure cannot be started due to the following situations:

- No unit procedure for the scanned device is available for processing at the current work center.
- For the scanned device, more than one unit procedure is available for processing at the current work center. This applies to parallel unit procedures on the same work center or to non-unique serial numbers for different products (dependent on the configuration of the barcode).

Confirm by Scanning (SR3071.4.2.5)

Operators can confirm a certain process step by scanning a barcode rather than tapping a button. Thus, navigation through a process or workflow can be done faster.

The barcode (e.g. 2D barcode) can be placed on the related button and scanned from the screen.

Server Heartbeat Check (SR3071.4.1.2)

➢ For recent changes, see revision history (page 89).

The system periodically runs a check of the required server communication. In case a communication error is detected, the system displays an appropriate message at each of the running Production Execution Clients.

The list of servers that are included in the heartbeat check can be configured by a system administrator. Per default, the check applies to the following servers:

- Electronic Batch Recording (EBR) server
  It controls the execution of EBR recipes and workflows and processes incoming messages from a Distributed Control System.
- Triggered Operation Management (TOM) server
  It manages event-triggered operations.
- Operation Execution (OE) server
  It controls the execution of server-run operations.
- Automation Integration (AI) server
  It controls the communication with automation-related systems.
- Status update (Transition server)
  It performs automatic, system-triggered status changes on objects, such as master recipes, master workflows, batches, orders, or workflows.

For configuration details, see chapter "Installing PharmaSuite Enterprise Edition on FactoryTalk ProductionCentre with JBoss" in "Technical Manual Installation - Enterprise Edition" [A6] (page 87).

Automatic Locking (SR3071.4.1.4)

The system detects user activity, including mouse movement, mouse clicks, and keyboard, touch, and barcode actions at a Production Execution Client. In case no user activity is detected for a certain amount of time, the system locks the client with the display of the **Change User** dialog in order to prevent unauthorized activity at the Production Execution Client. When a client is unlocked, the **User Access to Station Constraints (SR3071.4.8.3)** checks (page 13) and **User Change Constraints (SR3071.4.8.3.1)** checks (page 14) apply.

By default, automatic locking is enabled. It can be disabled. The inactivity timeout can be configured.

For configuration details, see chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A11] (page 87).

OE Server-specific Error Message (SR3071.4.1.3)

In case the system determines issues during processing that are related to the Operation Execution (OE) server or its running operations or phases, the system automatically broadcasts an error message and displays it at the running Production Execution Clients.

- Issues related to the execution of phases or operations result in error messages at the Production Execution Clients of the work centers of the unit procedure.
- Issues not related to a specific work center result in error messages at all running Production Execution Clients.

Processing issues that result in an error message at a Production Execution Client include:

- Phase-related issues that are caused by implementation errors of a phase.
- Phase-related issues that can be handled by the phase itself, but that are routed to the OE server error message mechanism by design.
- OE server-related issues that may occur while running the OE server, e.g. messaging, communication to other servers, or database connectivity.

Work-in-process Management (SR1094+)

The system supports work-in-process management.

Atomic transactions and traceability (SR1094.8)

All activities during process execution on the shop floor that affect the inventory trigger the recording of transaction log entries as defined in "Functional Requirement Specification Data Management" [A7] (page 87). At least this has to be performed for the following activities:

- creation of a batch,

creation of a sublot,

- quantity changes of a sublot, and
- relocation of a sublot.

Sufficient tracking information is recorded to assure the backward and forward traceability for batches and sublots, i.e. for a certain point in time it is possible to trace all inventory transactions that have been performed before and afterwards.

UoM conversion during execution (SR1094.47)

The system supports unit of measure conversion factors, so that different units of measure can be used.

Review of conversion factor (SR1094.47.1)

The batch record contains the original data that is related to quantities and conversion factors.

Review quantity of conversion factor (SR1094.47.1.1)

The system stores original values including units of measure, if applicable, that have been retrieved by user input or via interfaces. This is for documentation purposes.

Hierarchy of conversion factors (SR1094.47.2)

The system supports a hierarchy of conversion factors, starting at the most-specific conversion factor, if available:

- Material-specific conversion
- System-specific conversion

Set Initial Station (SR3071.4.3.2.1)

PharmaSuite provides a function to set the initial station with which the Production Execution Client is started.

If no station has been selected, in the Cockpit of the Production Execution Client only the actions of the **System** group are available.

GUI Components and Navigation for Pre-defined Workflows (SR3071.4.2+)

PharmaSuite provides the following GUI components and navigation capabilities for pre-defined workflows that are started from the Cockpit.

GUI Areas (SR3071.4.2.1)

➢ Applies to pre-defined workflows only.

The UI provides all information required for the current process step in a well-arranged way. It is separated into three different areas:

- Status bar (SR3071.4.2.1.3)
- Work area (SR3071.4.2.1.2)
- Navigation bar (SR3071.4.2.1.1)

Navigation Bar (SR3071.4.2.1.1)

- ➢ Applies to pre-defined workflows only.

The UI provides a navigation bar to navigate between the different workflow steps. Already closed steps can be viewed, but not changed anymore.

Alternative Workflow Steps (SR3071.4.2.1.1.2)

- ➢ Applies to pre-defined workflows only.

If the next workflow step is a fork, the system displays the alternative steps.
Fork means that there is more than one possible next step available.

Work Area (SR3071.4.2.1.2)

- ➢ Applies to pre-defined workflows only.

The UI provides a work area where required information for the current step is displayed or needed input can be performed.

Status Bar (SR3071.4.2.1.3)

- ➢ Applies to pre-defined workflows only.

The UI provides a status bar to display the current work center, the local time, information about the current step (step or activity name), and to give access to the help system. Other general functions, like cancel and switch to the list of workflows in the Cockpit are available, if required.

Session Handling (SR3071.4.2.2)

- ➢ Applies to pre-defined workflows only.

The system supports session handling and allows roll-back.

Chapter 3

Electronic Batch and Workflow Recording (SR1089+)

This section provides the requirements relevant to electronic batch and workflow recording:

Execution Framework - Basics (SR1089.1)

➢ For recent changes, see revision history (page 89).

The execution framework allows to execute control recipes and workflows within Production Execution Client.

This includes:

- Start the execution of unit procedures upon selection of the respective order step or workflow step.
  Order steps of the **Dispense** type are only available for selection at work centers of the **Dispense** type.
  See also the **Start Order Step (SR1094.9)** requirement (page 24).

The following status changes apply:
ORDER/WORKFLOW STEP
The system sets the order/workflow step status to **In process** (see **Status Management of Order Steps (SR1094.11)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87)).
UNIT PROCEDURE
The system sets the unit procedure status to **Running.**
ORDER/WORKFLOW
The system sets the order/workflow status to **In process** when the first order/workflow step of an order/workflow is started (see **Status Management**

of Orders (SR1084.4) in "Functional Requirement Specification Runtime Data Management" [A4] (page 87)).

- Start the execution of a one-click workflow upon selection of the respective workflow.

  The following status change applies:
  UNIT PROCEDURE
  The system sets the unit procedure status to **Running**.

- Execution of individual building blocks per S88-based recipe or workflow definition.
- Finish an order step or workflow step upon confirmation of the last phase. See also the **Finish Order Step (SR1094.10)** requirement (page 24).

  The following status changes apply:
  ORDER/WORKFLOW
  The system sets the order/workflow step status to **Finished** (see **Status Management of Order Steps (SR1094.11)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87)).
  UNIT PROCEDURE
  The system sets the unit procedure status to **Completed**.
  ORDER/WORKFLOW
  The system sets the order/workflow status to **Finished** when the last order/workflow step of a recipe/workflow path of an order/workflow is finished (see **Status Management of Orders (SR1084.4)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87)).

- Execution of reactivated order steps or workflow steps.

Batch Processing (SR1089.1.13)

The system supports the processing of batch-specific control recipes as an order-related process. The control recipe is based on a master recipe designed with Recipe Designer - Batch.

Selecting order steps to start batch processing (SR1089.1.13.1)

The system allows the operator to select one unit procedure to start batch processing.

The system provides a Selection Window with the following batch processing-specific attributes:

- Order step identifier
- Order step status
- Material identifier
- Material description
- [Planned] Quantity

- Target batch
- ERP start date
- ERP end date

The intelligent filter of the grid is applicable to the text values of any column.

In addition, startable unit procedures can be filtered by status.

Device Processing (SR1089.1.14)

The system supports the processing of device-specific control recipes as an order-related process. The control recipe is based on a master recipe designed with Recipe Designer - Device.

Selecting order steps to start device processing (SR1089.1.14.1)

The system allows the operator to select one unit procedure to start device processing.

The system provides a Selection Window with the following device processing-specific attributes:

- Serial number
- Batch identifier
- Material identifier
- Material description
- Order step identifier
- [Order step] Status
- Queued since

The intelligent filter of the grid is applicable to the text values of any column.

In addition, startable unit procedures can be filtered by status.

Workflow Processing (SR1089.1.15)

The system supports the processing of a workflow as a request-related process. The master workflow is designed with Workflow Designer. The request character is noticeable by the fact that the workflow has already been created and released in the Production Management Client prior to its execution in the Production Execution Client.

Selecting workflow steps to start workflow processing (SR1089.1.15.1)

The system allows the operator to select one unit procedure to start workflow processing.

The system provides a Selection Window with the following workflow processing-specific attributes:

- [Workflow] Processing name
- Workflow step [identifier (workflow, unit procedure)]
- [Workflow step] Status
- [Workflow] Type
- [Workflow] Short description
- Planned start [date]
- Planned end [date]

The intelligent filter of the grid is applicable to the text values of any column.

In addition, startable unit procedures can be filtered by status.

Start Order Step (SR1094.9)

The system allows the manufacturing operator to start an order step.

Finish Order Step (SR1094.10)

The system allows an operator to finish an order step.

> An order step is finished automatically after the operator completes the last operation of a unit procedure.

Consume sublots with last order step (SR1094.10.1)

In case a Dispensing order step is finished and no other order steps are following, the system can either automatically consume the generated target sublots during the order step finish or keep the target sublots including their given quantities.
The actual system behavior can be configured. The default configuration is set to automatic consumption of new target sublots.

Distributed Execution (SR1089.1.1)

Distributed execution is supported.
Process flow of S88 operations and phases is controlled on **client level** and the process flow of S88 unit procedures is controlled on **server level**.
For server-run operations, operations and phases run on the Operation Execution server (see **Server-run capability (SR3146.9.5.9.5)** in "Functional Requirement Specification Recipe and Workflow Management" [A2] (page 87)).

Unit Procedure Level Branching (SR1089.1.2)

For unit procedures, the system supports

- sequential execution and
- parallel execution.

Loops are not supported for unit procedures.

Operation Level Branching (SR1089.1.3)

For operations, the system supports

- sequential execution,
- parallel execution,
- XOR branching, and
- loops.

Phase Level Branching (SR1089.1.4)

For phases, the system supports

- sequential execution,
- parallel execution,
- XOR branching, and
- loops.

Information Flow during Execution (SR1089.1.5)

During execution, the system supports the information flow between phases, based on the configuration of input and output variables during recipe and workflow authoring.
The output value of a phase that has not been completed is Null.

Default preview at runtime (SR1089.1.5.1)

During execution, the system provides a default preview of a phase in case the phase returns a preview-related error (e.g. due to input parameters not being available).

Multiple phase instances within an operation (SR1089.1.5.2)

Multiple instances of a phase result from loops within an operation.

If the output variable of such a phase is referenced, the result is always the value of the output variable of the latest completed instance of that phase.

Multiple instances of operations (SR1089.1.5.4)

Multiple instances of operations result from loops within a unit procedure or from multiple runs of an **event-triggered operation (SR1089.6.2.1)** procedural element (page 36).

If the output variable of a phase of such an operation is referenced, the result is always the value of the output variable of this phase in the context of the operation with the highest instance count. This also applies to instances of event-triggered operation templates.

In case an event-triggered run has been canceled or an event-triggered operation template has been removed without any executed run, those instances are ignored, even if their instance count is the highest one.

If multiple instances of that phase exist within the given operation instance, still the **Multiple phase instances within an operation (SR1089.1.5.2)** feature (page 25) applies.

Multiple instances of unit procedures (SR1089.1.5.5)

Multiple instances of unit procedures result from reactivated unit procedures.

If the output variable of a phase within such a unit procedure is referenced, the result is always the value of the output variable of this phase in the context of the unit procedure with the highest instance count.

If multiple instances of that phase exist within an operation instance of that specific unit procedure instance, still the **Multiple phase instances within an operation (SR1089.1.5.2)** feature (page 25) applies.

Post-completion update of output variables (SR1089.1.5.3)

In case an output variable is related to a value that has been updated by means of a post-completion exception, the updated output variable is used as input for subsequent process parameter expressions and transition conditions. This only applies to phases and transitions that have not been activated yet while the update occurs.

One Operation at Multiple Work Centers (SR1089.1.6)

➢ Applies to the execution of control recipes only.

During execution, the system allows to process an order step of the **Dispense** type at multiple work centers of the **Dispense** type at the same time.

Operation Context (SR1089.1.7)

➢ Applies to the execution of control recipes only.

The system can manage specific context information on operation level and make it available to the related phases.

Campaign Execution (SR1089.1.8)

➢ Applies to the execution of control recipes only.

The system supports the execution of Dispense campaigns (Material processing).

Add material (SR1089.1.8.1)

The system provides a list of materials to the operator that are available for processing, allows to add materials to the campaign, and to remove materials from the campaign. The selection of materials is persistent.

The Selection Window provides the following material processing-specific attributes:

- Material identifier
- Material description
- Remaining quantity
- Allocated batch
- Order identifier
- Target batch
- Material (product) identifier
- Material (product) description
- [Work center] Cleaning [status]
- [Order step] Status

The intelligent filter of the grid is applicable to the text values of any column.

Execute unit procedure (SR1089.1.8.2)

During processing of materials within a campaign, the system executes the Dispense unit procedure of the related order. Only materials that belong to the campaign are visible and available for processing.

Update campaign (SR1089.1.8.3)

The system allows to add materials to or remove materials from an already started campaign.

Materials bound to different work centers (SR1089.1.8.4)

For materials that are already selected for a campaign or already identified at a different work center, the respective work center identifier is represented in the **Status** column of the material processing grid.

Execution is bound to work centers of Dispense type (SR1089.1.8.5)

The execution of Dispense campaigns (Material processing) is only possible at work centers of the **Dispense** type.

Parallel Operations on Different Stations (SR1089.1.9)

➢ Applies to the execution of control recipes only.

During execution and for a given work center, the system allows to process parallel operations on different stations at the same time.

Parallel Operations on Different Devices (SR1089.1.10)

➢ Applies to the execution of control recipes only.

During execution and for a given station, the system allows to process parallel operations on different devices at the same time.

One Device per Work Center of Dispense Type (SR1089.1.11)

During execution, the system allows only one device to be connected to a work center of the **Dispense** type.

System Recovery after Server Restart (SR1089.1.12)

➢ For recent changes, see revision history (page 89).

Upon restart of a PharmaSuite server component, the system is able to recover and to retrieve the actual system status information from the database, so that the execution can be continued.

This applies to the following servers:

- Electronic Batch Recording (EBR) server
  It controls the execution of EBR recipes and workflows and processes incoming messages from a Distributed Control System.
- Triggered Operation Management (TOM) server
  It manages event-triggered operations.
- Operation Execution (OE) server
  It controls the execution of server-run operations.

Workflow Specifics (SR1089.7+)

Validity check for one-click startable workflows (SR3071.4.3.2.2)

Only workflows can be started for execution that are in the **Valid** status after they have been approved in the Workflow Designer.

See also **Version Control (SR3146.4.5)** in "Functional Requirement Specification Recipe and Workflow Management" [A2] (page 87).

Start and append workflows (SR1089.7.1)

➢ For recent changes, see revision history (page 89).

The system allows to define an access privilege for the protection of master recipes and master workflows from unauthorized access. Subsequently, it shall only allow to select a workflow for starting and an order for appending whose access privilege matches the access privilege of the logged-in user.

The system allows the operator to start and optionally append a workflow to a specific order step of a batch- or device-specific order.

A started workflow can be appended to a running batch- or device-specific unit procedure if:

- the **Appendable during processing** attribute is set,
- the workflow is not in the **Canceled** status, and
- the unit procedure is visible in the Cockpit of the current device.

If the workflow cannot be appended, the system displays an appropriate message.

Appended workflows are considered as **Production-relevant**. This affects the **Purge workflow (SR1085.2.11)** operation, see "Functional Requirement Specification Runtime Data Management" [A4] (page 87).

Exported workflows cannot be appended to an order according to the **Export Workflow for Archive (SR1085.2.10)** operation, see "Functional Requirement Specification Runtime Data Management" [A4] (page 87).

Select unit procedure to append workflow (SR1089.7.1.1)

➢ For recent changes, see revision history (page 89).

The system allows to define an access privilege for the protection of master recipes from unauthorized access. Subsequently, it shall only allow to select a unit procedure of an order whose access privilege matches the access privilege of the logged-in user.

If more than one batch- or device-specific unit procedure is available for the workflow to be appended to, the system allows the operator to select a specific running unit procedure.

The system provides a Selection Window with the following unit procedure-specific attributes:

- Unit procedure name
- Order identifier
- Actual start date

The intelligent filter of the grid is applicable to the text values of any column.

Append workflow-specific exception at workflow start (SR1089.7.1.2)

The append operation is associated with an append workflow-specific exception on unit procedure level. The exception requests a pre-defined signature according to the **PEC_Signature_AppendWorkflowsToOrder** access privilege for recording. In the default configuration, the exception is deactivated.
The append workflow-specific exception is an exception of the **user-triggered (SR1079.1.2)** category (page 69).
Upon confirmation of an append workflow-specific exception, the exception is recorded for the unit procedure to which the workflow is appended. The risk level of the exception can be configured.

For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).

For details related to the activation of the exception, see chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A11] (page 87).

For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Resuming the Execution (SR1089.2)

The system allows to resume the execution of a control recipe or workflow where it has been left off. This applies to the unit procedure and the operation levels.

Unit Procedure-related Capabilities (SR1089.8+)

Detaching a Unit Procedure (SR1089.8.4+)

Detach unit procedure (SR1089.8.4.1)

The system allows an operator to detach a running unit procedure from its work center and station. The availability of the detach action during execution is defined in the master recipe.

A unit procedure can only be detached if

- it is not in the **Paused** status (page 33),
- all operations of the unit procedure that are running on a client are either completed or detached, and
- no scale is bound in the context of the unit procedure (applies to Inline Weighing and Output Weighing).

If the unit procedure cannot be detached, the system displays an appropriate message.

A detached unit procedure and its operations are no longer visible in the **Cockpit (SR3071.4.1.1)** view (page 6).

Detach-specific exception (SR1089.8.4.2)

The detach operation for a unit procedure can be associated with a detach-specific exception on unit procedure level. The availability of the exception during execution is defined in the master recipe.
The detach-specific exception is an exception of the **user-triggered (SR1079.1.2)** category (page 69).

For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).

Recording detach-specific exceptions (SR1089.8.4.2.1)

Upon confirmation of a detach-specific exception, the exception is only recorded for the unit procedure.

Resume comment of detach-specific exceptions (SR1089.8.4.2.2)

Upon resume of a formerly detached unit procedure, the system automatically adds a comment to the related detach-specific exception. This comment records the timestamp of the resume action.

Context-related Actions (SR1089.8.1)

The system provides the following actions related to unit procedures in an Actions dialog:

- **Pause (SR1089.8.3)** action (page 33)
- **Manage exceptions (SR1079+)** action (page 67)
  Add user-defined exceptions (SR1079.1.1) action (page 67)
- **Abort and reactivate (SR1089.8.5)** action (page 34)
- **Append workflow (SR1089.8.2)** action (page 31)

Appending Workflows to a Unit Procedure (SR1089.8.2)

➢ For recent changes, see revision history (page 89).

The system allows to define an access privilege for the protection of master workflows from unauthorized access. Subsequently, it shall only allow to select a workflow whose access privilege matches the access privilege of the logged-in user.

The system allows the operator to select and append workflows to a specific order step of a batch- or device-specific order.

A workflow can be appended to a running batch- or device-specific unit procedure if the **Appendable during processing** attribute is set, the workflow has been started, and the workflow is not in the **Canceled** status.

If the workflow cannot be appended, the system displays an appropriate message.

> Appended workflows are considered as **Production-relevant**. This affects the **Purge workflow (SR1085.2.11)** operation, see "Functional Requirement Specification Runtime Data Management" [A4] (page 87).
>
> Exported workflows cannot be appended to an order according to the **Export Workflow for Archive (SR1085.2.10)** operation, see "Functional Requirement Specification Runtime Data Management" [A4] (page 87).

Selecting workflow to append to unit procedure (SR1089.8.2.1)

➢ For recent changes, see revision history (page 89).

The system allows to define an access privilege for the protection of master workflows from unauthorized access. Subsequently, it shall only allow to select a workflow whose access privilege matches the access privilege of the logged-in user.

The system allows the operator to select one or more workflow processes to be appended to the current unit procedure of a batch- or device-specific order.

The system provides a Selection Window with the following workflow-specific attributes:

- [Workflow] Processing name
- Workflow identifier, [Workflow] Status
- Master workflow identifier
- Master workflow short description
- Executed at work center
- Actual start date
- Actual finish date

Workflows already appended to the current unit procedure are not selectable.

The following options allow to control the grid rows to be displayed:

- Intelligent filter of the grid is applicable to the text values of any column.
  Default: no filter
- Filter on time periods since the workflow has been started
  Default: Last eight hours
- Filter on the workflow step status
  Default: Finished

- Filter if workflow run on the current work center only
  Default: Yes

Append workflow-specific exception during unit procedure processing (SR1089.8.2.2)

The append operation is associated with an append workflow-specific exception on unit procedure level. The exception requests a pre-defined signature according to the **PEC_Signature_AppendWorkflowsToOrder** access privilege for recording. In the default configuration, the exception is deactivated.
The append workflow-specific exception is an exception of the **user-triggered (SR1079.1.2)** category (page 69).
Upon confirmation of an append workflow-specific exception, the exception is recorded for the unit procedure to which the workflows are appended. The risk level of the exception can be configured.

> For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).
>
> For details related to the activation of the exception, see chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A11] (page 87).
>
> For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Pausing a Unit Procedure (SR1089.8.3)

The system allows the operator to pause and resume a paused unit procedure. The availability of the Pause-enabled capability of a unit procedure during execution is defined in the master recipe.

> Only pause-aware phases are affected. Other phases can still be processed.

Graphical representation of a paused unit procedure (SR1089.8.3.1)

The **Paused** status is represented in the notification bar of the **header bar (SR3071.4.2.4.1)** area (page 4) and in the **Cockpit (SR3071.4.1.1)** view (page 6) by a notification marker to the left of the unit procedure.

Pause unit procedure-specific exception (SR1089.8.3.2)

The pause operation can be associated with a pause unit procedure-specific exception on unit procedure level. The availability of the exception during execution is defined in the master recipe.
The pause unit procedure-specific exception is an exception of the **user-triggered**

(SR1079.1.2) category (page 69).
Upon confirmation of a pause unit procedure-specific exception, the exception is recorded.

For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).

Resume comment of pause unit procedure-specific exceptions (SR1089.8.3.2.1)

Upon resume of a formerly paused unit procedure, the system automatically adds a comment to the related pause unit procedure-specific exception. This comment records the timestamp of the resume action.

Reactivating a Unit Procedure (SR1089.8.5)

The system allows the operator to abort and reactivate a running unit procedure. The availability of the "abort and reactivate" action during execution is defined in the master recipe.

If an operation of the unit procedure is running at another station or work center, the system displays an appropriate message at each of the related Production Execution Clients.
For reactivating a unit procedure, similar preconditions apply as for the **detaching a unit procedure (SR01089.8.4.1)** function (page 30).

- The unit procedure is not in the **Paused** status (page 33),
- uninterruptable operations of the unit procedure that are running on a client are either completed or detached, and
- no scale is bound in the context of the unit procedure (applies to Inline Weighing and Output Weighing).

If the preconditions are violated, the system displays an appropriate message.

A reactivated unit procedure and its operations are no longer visible in the **Cockpit (SR3071.4.1.1)** view (page 6).

Reactivate unit procedure-specific exception (SR1089.8.5.1)

The "abort and reactivate" operation can be associated with a specific exception on unit procedure level. The availability of the exception during execution is defined in the master recipe.
The reactivate unit procedure-specific exception is an exception of the **user-triggered (SR1079.1.2)** category (page 69).

The system automatically records an "abort unit procedure"-specific exception for all active phases of the unit procedure. The risk level of the exception can be configured.

For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).

For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Recording reactivate unit procedure-specific exception (SR1089.8.5.2)

Upon confirmation of a reactivate unit procedure-specific exception, the exception is recorded for the running unit procedure and the unit procedure is aborted.

Operation-related Capabilities (SR1089.6+)

The system supports the definition and configuration of certain operation-related capabilities within Recipe and Workflow Designer, see [A2] (page 87). This section describes the respective system behavior during execution.

Detaching an Operation (SR1089.6.1+)

Detach operation (SR1089.6.1.1)

The system allows an operator to detach a running operation from its device and station. The availability of the detach action during execution is defined in the master recipe.

A detached operation is no longer flagged as running operation with the **indicator for running operations (SR3071.4.1.1.5)** feature (page 8).

Detach Inline Weighing operation (SR1089.6.1.1.1)

An Inline Weighing operation can only be detached if the operation can be held, i.e. no scale is bound to the operation.

If the operation cannot be detached, the system displays an appropriate message.

Detach-specific exception (SR1089.6.1.2)

The detach operation for an operation can be associated with a detach-specific exception on operation level. The availability of the exception during execution is defined in the master recipe.
The detach-specific exception is an exception of the **user-triggered (SR1079.1.2)** category (page 69).

For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).

Recording detach-specific exceptions (SR1089.6.1.2.1)

Upon confirmation of a detach-specific exception, the exception is recorded for each of the running phases of that operation and the operation is detached.

Restart comment of detach-specific exceptions (SR1089.6.1.2.2)

Upon restart of a formerly detached operation, the system automatically adds a comment to the related detach-specific exception. This comment records the timestamp of the restart.

Triggering an Operation by Events (SR1089.6.2+)

Event-triggered operation (ETO) (SR1089.6.2.1)

The system supports operations that are triggered by events during execution. The availability of the Event-triggered capability of an operation during execution is defined in the master recipe.

Representation of event-triggered operations (SR1089.6.2.2)

Initially, an event-triggered operation is a **template** displayed as an operation that is not startable. The template information consists of the operation's identifier and description. When templates are looped, the system appends a count to their operation identifiers according to the **loop counter (SR3071.4.3.2)** feature (page 10).

As soon as a **run** is created based on a template, the representation changes to that of an idle operation. The run information consists of the operation's identifier and timing-specific information of the current run.
Several runs of one template can be created. The system appends a count to the operation identifier starting with the first run.

The system provides the following actions related to event-triggered operations (ETO-specific actions) in an Actions dialog:

- **Create a new run (SR1089.6.2.3)** action (page 37)
- **Cancel a run (SR1089.6.2.4)** action (page 37)
- **Cancel all runs (SR1089.6.2.5)** action (page 37)
- **Remove a template (SR1089.6.2.8)** action (page 37)

> The timing-specific information of a run supports the prioritization of operators' activities.

Create a new run (SR1089.6.2.3)

A new run is created based on the current template. The related operation is idle.

When the first run is created, the representation of the template changes to the representation of an idle operation. Each additional run is positioned above the existing runs.

When the last run of a template is completed, the representation of the run changes back to the representation of the template.

Create a new run-specific exceptions (SR1089.6.2.3.1)

➢ For recent changes, see revision history (page 89).

The system automatically records a "create new run"-specific system-triggered exception when the Triggered Operation Management (TOM) server is not available when a new run is to be created. The risk level of the exception can be configured.

> For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).
>
> For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Cancel a run (SR1089.6.2.4)

A run can only be canceled as long as the related operation is idle. A canceled run is removed from all stations.

When the last run of a template is canceled, the representation of the run changes to the representation of the template.

Cancel all runs (SR1089.6.2.5)

Multiple active runs of one template can be canceled at once. For each individual run, the system follows the same behavior as during the cancelation of a single run with the **Cancel a Run (SR1089.6.2.4)** action (page 37).

> If an exception has been defined for the **Cancel all runs** action, the system records an exception for each run, however the operator signs only one exception.

Remove a template (SR1089.6.2.8)

➢ For recent changes, see revision history (page 89).

A template can only be removed if no runs are available (active or running).

When a template is removed, its representation disappears from the Cockpit at all related stations.

If the Operation Execution (OE) server, the Triggered Operation Management (TOM) server, or the message broker do not run, the template cannot be removed. The system displays an appropriate message.

ETO-specific exception (SR1089.6.2.9)

Each of the ETO-specific actions can be associated with a related exception on operation level. The availability of the exception during execution is defined in the master recipe. The ETO-specific exception is an exception of the **user-triggered (SR1079.1.2)** category (page 69).

If an operation expires while an exception related to the **Create a new run**, **Cancel a run**, or **Cancel all runs** actions is being recorded, the system displays an appropriate message and then returns to the Cockpit.

For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).

Recording ETO-specific exceptions (SR1089.6.2.9.1)

Upon confirmation of an ETO-specific exception, the exception is recorded for the ETO operation. This applies to templates and runs.

Recording ETO-specific exceptions - Actions not completed (SR1089.6.2.9.2)

In case an ETO action-specific exception has been recorded, but the action cannot be completed by the system, the system automatically adds a comment to the related ETO action-specific exception. This applies to the following circumstances

- Conflicts with ETO-related actions of other clients.
- Issues related to the connectivity to the TOM server.

Escalation Management of Event-triggered Operations (SR1089.6.3+)

Escalation stages (SR1089.6.3.1)

The system supports four escalation stages of a run of an event-triggered operation:

1. Run due
   The run is idle and has become due since it was triggered.
2. Run due reminder
   The run is idle and its reminder alerts the operator.
3. Run overdue
   The run is idle and overdue.

4. Run expired
   The run is expired and disappears from the list of **Startable** operations within the Cockpit.

Graphical representation of escalation stages (SR1089.6.3.2)

The escalation stages are represented in the notification bar of the **header bar (SR3071.4.2.4.1)** area (page 4) and in the **Cockpit (SR3071.4.1.1)** view (page 6) by notification markers to the left of the operation.

In the notification bar, the system displays a specific notification tab with the current number of runs for each of the following stages:

- Run due,
- Run due reminder,
- Run overdue.

The system displays a notification marker in the Cockpit for each run of the following stages:

- Run due reminder,
- Run overdue.

Escalation-specific exceptions (SR1089.6.3.4)

Each of the following escalation stages can be associated with a specific exception on operation level. The availability of the exception during execution is defined in the master recipe.
The escalation-specific exceptions are exceptions of the **system-triggered (SR1079.1.4)** category (page 70) (see **Implicit Capability-related Exceptions (SR1079.1.4.2)** for details (page 72)). They are recorded automatically by the system without any user interaction.

> For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).

Server-side Running Operations (SR1089.6.4+)

Server-run definition (SR1089.6.4.1)

The system supports operations that run on a server. The availability of the Server-run capability of an operation is defined in the master recipe.

Server-side running operations are not visible in any view of the Production Execution Client.

Operation Execution (OE) server-run-specific exceptions (SR1089.6.4.2)

➢ For recent changes, see revision history (page 89).

The system supports the recording of a server-run-specific exception in case an operation that runs on an OE server is restarted. The exception is added on operation level.

The OE server-run-specific exceptions are exceptions of the **system-triggered (SR1079.1.4)** category (page 70) (see **Implicit Capability-related Exceptions (SR1079.1.4.2)** for details (page 72)). They are recorded automatically by the system without any user interaction. The risk level of the exception can be configured.

> For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).
>
> For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Triggered Operation Management (TOM) server-run-specific exceptions (SR1089.6.4.3)

➢ For recent changes, see revision history (page 89).

The system supports the recording of a server-run-specific exception in case a template that runs on a TOM server is restarted. The exception is added on unit procedure level.

The TOM server-run-specific exceptions are exceptions of the **system-triggered (SR1079.1.4)** category (page 70) (see **Implicit Capability-related Exceptions (SR1079.1.4.2)** for details (page 72)). They are recorded automatically by the system without any user interaction. The risk level of the exception can be configured.

> For details related to exceptions in general, see the **Exception Management (SR1079+)** feature (page 67).
>
> For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Auto-startable Operations (SR1089.6.5)

The system supports operations that can be started automatically rather than being started manually in the **Cockpit (SR3071.4.1.1)** view (page 6) of the Production Execution Client. In this case, the Production Execution Client directly switches to the **Execution Window (SR3071.4.2.4.2)** view (page 12) with its running phases.

The availability of the Auto-startable capability of an operation is defined in the master recipe.

Auto-start constraints (SR1089.6.5.1)

An operation with the Auto-startable capability is automatically started or resumed whenever one of the following scenarios applies:

- An order step (unit procedure) is started for the first time and the auto-startable operation is the first operation of this unit procedure.
  This also applies to reactivated order steps (unit procedures).
- A unit procedure is running and the predecessor operation of the auto-startable operation is completed.
- A unit procedure was detached while the auto-startable operation was running and the related order step is now restarted.

An operation with the Auto-startable capability is not automatically started in case at least one of the following constraints applies:

- The unit procedure runs on multiple stations or devices.
- No Production Execution Client is running at a required station at the point of time when the auto-start would have been triggered.
- The **Register at Station** dialog is open.
- The **Change User** dialog is open.
- The context dialog of unit procedures or operations is open.

Specific API Support for Building Blocks (SR1089.3+)

The following dedicated APIs are available to support building blocks.

API - Material Identification (SR1089.3.1+)

Material identification is supported in the context of a control recipe.

Checks (SR1089.3.1.1)

Building blocks can control material-related checks and exception handling.

API - Material Accounting (SR1089.3.2+)

Material accounting is supported in the context of a control recipe.

Types (SR1089.3.2.1)

Building blocks can record the consumption of sampled and wasted quantities.

Correction (SR1089.3.2.2)

Building blocks can record a correction of sampled and wasted quantities.

Release sublot (SR10893.2.3)

Building blocks can release a sublot without any consumption activities at that time.

API - Data Recording (SR1089.3.3+)

Multiple records (SR1089.3.3.1)

Building blocks can record and retrieve multiple data records. The number of data records is defined dynamically during execution.

Support of runtime properties (SR1089.3.3.2)

During execution, a phase can write and read runtime property-related data of an equipment entity.

API - Status Handling of Phases (SR1089.3.4+)

Phase UI status: Visible/invisible (SR1089.3.4.1)

System events can determine when the UI status of a phase changes to **Visible** or **Invisible**. This applies to all UI views of a phase, which are related to execution, exceptions, or post-completion actions.

Phase S88 status: Hold (SR1089.3.4.2)

A system event recognizes when the S88 status of a phase changes to **Hold**. The event occurs when the Production Execution Client is shut down.

API - Usage Control of Equipment Entities (SR1089.3.5+)

Use cases (SR1089.3.5.1)

The system can control the usage of an equipment entity in the context of an order or workflow.
This includes:

- Monitoring the actual usage.
- Preventing a concurrent usage.
- Providing mechanisms to make an equipment entity available again after its usage.

Status graph (SR1089.3.5.2)

The system supports the following statuses and transitions for equipment entities as part of the usage control:

- **Available**: Equipment entity is not in use.
- **Identified**: Equipment entity is reserved for further usage in its context.
- **Bound**: Equipment is bound to its context.

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
1.1 - Bound » Available	---	Automatic transition (Release).
1.2 - Identified » Available	---	Automatic transition (Release).
2.1 - Available » Identified	---	Transition triggered by equipment identification (Identify).
3.1 - Identified » Bound	---	Automatic transition (Bind).

The diagram illustrates the **S88EquipmentBinding** flexible state model.

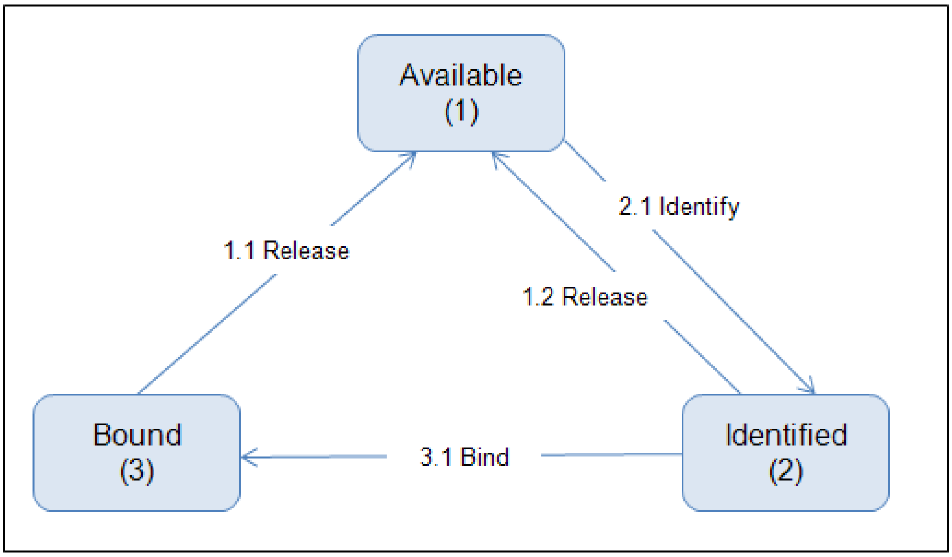

Figure 1: Usage control of equipment entity - status graph

Support of rule validation (SR1089.3.5.3)

During execution, a phase can validate equipment requirement-specific rules. This includes rules that are defined on the basis of property types or flexible rules that have been defined within the Expression editor.

API - Status Handling of FSM Properties of Equipment Entities (SR1089.3.6+)

FSM property status management (SR1089.3.6.1)

Building blocks can manage the status of an FSM property of a specific equipment entity. This includes the following API calls:

- Apply transition.
- Check for available transitions.
- Force status.

API - Recording of Equipment Entity Logbook (SR1089.3.7+)

Usage control (SR1089.3.7.1)

If a logbook is maintained for a specific equipment entity, the system tracks the following events related to usage control in the logbook:

- All transitions (Identify, Bind, Release).
- Failure of identification, if entity is already in the **Identified** status.

FSM property status change (SR1089.3.7.2)

If a logbook is maintained for a specific equipment entity, the system tracks the following events related to FSM property status changes in the logbook:

- Apply transition.
- Force status.

Phase information and signature (SR1089.3.7.3)

If a logbook is maintained for a specific equipment entity, a phase can record a specific information string in the logbook. The following information-related categories are supported:

- Manual: for manual logbook entries
- Denial: identification/binding of equipment entities has been denied
- Phase: recording of other phase-specific events

The API call also supports the recording of related signature data that might be provided by the phase.

> Customer-specific categories can be defined by a system integrator and can be used when a phase creates the related logbook entry.

API - Equipment Entity Logbook Data Retrieval (SR1089.3.9)

If a logbook is maintained for a specific equipment entity, the system allows to retrieve specific data that was recorded in the logbook (e.g. last product material).

The API supports the meaning of **last** with respect to the last binding action for this specific equipment entity. Example: **Last product material** returns the product material of the last order for which this specific equipment was used (i.e. bound and released), which is documented by the **Bound** binding status.

API - Context Information for Phase Dialogs (SR1089.3.8+)

System-triggered exception context (SR1089.3.8.1)

System-triggered exception dialogs of phases display the trigger timestamp and the identifiers of the operation and phase that triggered the exception.

Error, warning, and information message context (SR1089.3.8.2)

Error, warning, or information message dialogs of phases display the trigger timestamp and the identifiers of the operation and phase that triggered the dialog.

API - System-triggered Exceptions without User Interaction (SR1089.3.10)

The system supports system-triggered exceptions that can be triggered by a phase and do not require any user interaction.

Per phase-related business logic, the exception is recorded automatically by the system without any user interaction.

API - Generating an Equipment Entity (SR1089.3.11)

Based on a template equipment entity, the system can generate a defined number of equipment entities according to the data defined with the template.

The generation includes the following steps:

- Generate new entities.
- Set status of newly generated entities.

For technical details, see chapter "Handling S88 Equipment-related Features" in "Technical Manual Developing System Building Blocks" [A12] (page 87).

Number of entities to be generated (SR1089.3.11.1)

The number of entities to be generated can be configured.

Barcode and identifier (SR1089.3.11.2)

The system uses the following template-specific attributes to build the barcode and identifier of the generated entity:

- **Barcode prefix**:
  If the barcode prefix source is set to **Template-specific**, the system uses the value of the **Barcode prefix character** attribute. The **Barcode prefix character** attribute is not mandatory. If no value is defined, the barcode will be identical to the identifier.
  If the barcode prefix source is set to **From configuration key**, the system uses the value of the corresponding configuration key.
- **Identifier prefix**: If no value is defined, the system uses the value of the corresponding configuration key.
- **Sequencer length**: Defines the number of digits to be used for the count of generated entities that succeed the identifier prefix. The system uses leading zeros to fill up the digits in order to have barcodes with a constant length. If no value is defined, the system uses the value of the corresponding configuration key.

Example:
Barcode prefix character = §
Identifier prefix = EQ
Sequencer length = 5
The first of the generated entities has the identifier **EQ00001** and the barcode **§EQ00001**.

In case that one of the values (identifier, barcode) is already used by an equipment entity, the entity generation fails and the API call ends.

Several template entities can share the same sequencer. The sequencer is unique per identifier prefix.

Data of a generated entity (SR1089.3.11.3)

The data of a template entity is copied unchanged to the newly generated entity with the following exceptions:

- Template-specific **Entity Barcode** attributes are not copied.
- The **Template used** attribute is set to the identifier of the used template entity.
- Runtime property values are not copied. They are initialized with the value that is set when a property has been added to an entity.
- The statuses of the assigned equipment graphs are set to their initial statuses.
- Entity barcode and identifier are generated according to the **Barcode and Identifier (SR1089.3.11.2)** requirement (page 46).

Status of generated entity (SR1089.3.11.4)

The status of the generated entity can explicitly be set to the status of the template entity. If the status is not set explicitly, the status of the generated entity is set to **Draft**.

Allowed template entity status (SR1089.3.11.5)

Template entities in the **Approved** status or any lower status are allowed to be used for the generation of equipment entities.

Return values (SR1089.3.11.6)

The API call returns the generated entity objects.

API - Printing an Equipment Entity Label (SR1089.3.12+)

> For technical details, see chapter "Changing or Adding New Labels" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Label layout (SR1089.3.12.1)

The system uses the label layout specified with the **Label layout** attribute of the entity. If no label layout has been defined or if the defined label layout does not exist as a report design, the system uses the default layout as defined in the system configuration.

Printer (SR1089.3.12.2)

The printer to be used can be defined.

- If the defined printer does not exist, the system reports an error.
- If no printer has been defined, the system uses the default label printer as configured in the system configuration. If no printer has been configured in the system configuration, the system uses the default printer of the work station.

API - Generating an Equipment Entity of the Hybrid Equipment Type (SR1089.3.13)

Based on a sublot and a related template equipment entity, the system can generate a defined number of equipment entities of a defined quantity per entity from the sublot and according to the data defined with the template. The system uses the template entity that is assigned to the material of the sublot.

The generation includes the following steps:

- Perform material-related checks against the identified sublot.
- Perform an implicit inventory correction.

- Retrieve the template entity from the material of the sublot.
- Generate new entities.
- Consume or split sublot.
- Fill entity's runtime properties of the **Base Sublot (RS)** and **Current Sublot (RS)** purposes. The system updates the logbook accordingly (if maintained).
  It is assumed that a sublot identifier is unique within the system.
- Set status of newly generated entities.

For technical details, see chapter "Handling S88 Equipment-related Features" in "Technical Manual Developing System Building Blocks" [A12] (page 87).

Sublot identification checks (SR1089.3.13.1)

The system allows to configure which checks are performed against the sublot, the material of the sublot, or the batch of the sublot. The checks are defined in the check suite for sublot identification.
The check suite can consist of checks configured by phase building block parameters and/or non-adjustable checks.

The following checks are available and can be added to the check suite:

- Checks that are configured with a phase building block parameter and which require further actions (e.g. creating a system-triggered exception) if they fail:
  - Sublot batch minimum quality status
  - Sublot batch expiry date (with minimum time to expire)
  - Sublot batch retest date (with minimum time to retest)
- Checks that are not adjustable and cause the execution to be aborted if they fail:
  - Sublot blocked by other process
  - Identified sublot logically deleted
  - UoM of given quantity is not convertible to UoM of sublot quantity

The system allows to integrate further checks.

For details, see chapter "Handling S88 Equipment-related Features" in "Technical Manual Developing System Building Blocks" [A12] (page 87).

A failed check prevents the execution of further actions of the method. In case an exception is recorded for the failed check, the signature and the related checks can be provided to a second API call. The second call ignores all check-related errors that were already signed. However, the execution is aborted when a different check fails.

Implicit inventory correction (SR1089.3.13.11)

To determine the quantity to be consumed from the base sublot, the system multiplies the number of entities to be generated by the quantity to be consumed per entity.
In case the quantity of the base sublot is insufficient, the system behavior can be defined: either the API call returns an error or, before the consumption execution, the quantity is increased by the difference between required and available quantity.

Template entity to be used (SR1089.3.13.2)

The system retrieves the template entity to be used for the generation of a new entity from the material of the identified sublot. If there is no template entity defined for the material, the system reports an error.

Number of entities to be generated (SR1089.3.13.3)

The number of entities to be generated can be configured.

Each entity generation, including its sublot-related actions, is done within an own transaction.

Reason for generating hybrid entities (SR1089.3.13.4)

An entity of the **Hybrid** equipment type can be generated and used for two different reasons: **Split** or **Consume immediately**.

In both cases, a sublot must have been identified. For each generated entity, it is possible to define the quantity (and unit of measure) that has to be subtracted from the base sublot.

In case of **Split**, the consumption does not happen when the entity is generated. The sublot generated by the split operation is consumed later when the entity is used within an order context.

In case of **Consume immediately**, the defined quantity is immediately consumed from the base sublot.
If the base sublot is empty afterwards, it is marked as **logically deleted**.

Transaction history subtype for consumption (SR1089.3.13.5)

The transaction history subtype that is documented with the quantity changes of the base and current sublots can be defined as any value of the editable *TransactionSubtype* choice list.

Barcode and identifier (SR1089.3.13.6)

For equipment entities of the **Hybrid** equipment type, the same algorithm applies as for non-hybrid equipment entities. For details, please refer to the **Barcode and Identifier (SR1089.3.11.2)** requirement (page 46).

Allowed template entity status (SR1089.3.13.9)

Template entities in the **Approved** status or any lower status are allowed to be used for the generation of equipment entities.

Data of generated entity (SR1089.3.13.7)

The data of a template entity is copied unchanged to the newly generated entity with the following exceptions:

- Template-specific **Entity Barcode** attributes are not copied.
- The **Template used** attribute is set to the identifier of the used template entity.
- Runtime property values are not copied. They are initialized with the value that is set when a property has been added to an entity.
- The statuses of the assigned equipment graphs are set to their initial statuses.
- Entity barcode and identifier are generated according to the **Barcode and Identifier (SR1089.3.13.6)** requirement (page 49).
- The entity's property of the **Base Sublot (RS)** purpose is filled with the sublot identifier of the identified sublot.
- Only if the **Reason for Generating (SR1089.3.13.4)** the hybrid entity (page 49) is **Split**: The entity's property of the **Current Sublot (RS)** purpose is filled with the sublot identifier of the new entity that was split from the identified sublot.

Status of generated entity (SR1089.3.13.8)

The status of the generated entity is set to the status of the template entity.

Return values (SR1089.3.13.10)

The API call returns the generated entity objects and, if applicable, exceptions, warnings, and errors.

API - Identifying an Equipment Entity of the Hybrid Equipment Type (SR1089.3.14)

When identifying an equipment entity of the **Hybrid** equipment type, the system performs the regular equipment identification checks and actions. Additionally, further checks, a material identification, and inventory operations on the linked sublots can be performed.

Failed checks, material identification, or sublot operations prevent the binding of the equipment entity. If the entity belongs to an equipment entity group, the entire group is not bound.

If the entire identification has been performed successfully or all exceptions have been signed, the material and inventory actions are persistent and cannot be undone.

For technical details, see chapter "Handling S88 Equipment-related Features" in "Technical Manual Developing System Building Blocks" [A12] (page 87).

Current sublot identification checks (SR1089.3.14.1)

The system allows to configure which checks are performed against the sublot, the material of the sublot, or the batch of the sublot whose sublot identifier is stored in the entity's property of the **Current Sublot (RS)** purpose. The checks are defined in the check suite for sublot identification.
The checks on batch level are performed, even if the referenced sublot is already marked as **logically deleted**.
The check suite can consist of checks configured by phase building block parameters and/or non-adjustable checks.

The following checks are available and can be added to the check suite:

- Checks that are configured with a phase building block parameter and which require further actions (e.g. creating a system-triggered exception) if they fail:
  - Sublot batch minimum quality status
  - Sublot batch expiry date (with minimum time to expire)
  - Sublot batch retest date (with minimum time to retest)
  - Batch allocation check
  - **Material parameter quantity (SR1089.3.14.4)** check (page 53)
- Checks that are not adjustable and cause the execution to be aborted if they fail:
  - Sublot blocked by other process
  - Identified sublot already logically deleted
  - MFC-relevant material parameter exists for an identified sublot

The system allows to integrate further checks.

For technical details, see chapter "Handling S88 Equipment-related Features" in "Technical Manual Developing System Building Blocks" [A12] (page 87).

A failed check prevents the execution of further actions of the method. In case an exception is recorded for the failed check, the signature and the related checks can be provided to a second API call. The second call ignores all check-related errors that were already signed. However, the execution is aborted when a different check fails.

Base sublot identification checks (SR1089.3.14.2)

These checks are optional.
The system allows to configure which checks are performed against the batch of the sublot whose sublot identifier is stored in the entity's property of the **Base Sublot (RS)** purpose. The checks are defined in the check suite for sublot identification.
The checks are only performed if the entity's property of the **Current Sublot (RS)** purpose is empty.
The checks on batch level are performed, even if the referenced sublot is already marked as **logically deleted**.
The check suite can consist of checks configured by phase building block parameters and/or non-adjustable checks. The check suite can differ from the check suite of the **Current Sublot Identification (SR1089.3.14.1)** checks (page 51). The phase building block parameters are assumed to be equal.

The following checks are available and can be added to the check suite:
They are configured with a phase building block parameter.

- Sublot batch minimum quality status
- Sublot batch expiry date (with minimum time to expire)
- Sublot batch retest date (with minimum time to retest)

The system allows to integrate further checks.

For technical details, see chapter "Handling S88 Equipment-related Features" in "Technical Manual Developing System Building Blocks" [A12] (page 87).

A failed check prevents the execution of further actions of the method. In case an exception is recorded for the failed check, the signature and the related checks can be provided to a second API call. The second call ignores all check-related errors that were already signed. However, the execution is aborted when a different check fails.

Identification and consumption of current sublot (SR1089.3.14.3)

For the current sublot, the system supports that the identification and consumption of the sublot can be skipped. This can be configured.

In case the entity's property of the **Current Sublot (RS)** purpose is not empty and matches a material parameter, the referenced sublot will be identified and completely consumed. The consumed sublot is marked as **logically deleted**.

It is an error case if the sublot does not have a matching material parameter. This situation is already covered by the configurable check suite and the **MFC-relevant material parameter exists for an identified sublot** check (see **Current Sublot Identification (SR1089.3.14.1)** checks (page 51)).

If the sublot material does not match one material parameter unambiguously, the identification fails.

Material parameter quantity check (SR1089.3.14.4)

The system checks that the planned quantity of all MFC-relevant material parameters is reached. The check is only performed if a planned quantity is defined.

In case absolute and/or relative tolerances have been defined for a material parameter, the lower value is used to check if the consumed quantity is within the allowed range. A violation of the planned quantity generates an exception.

Return values (SR1089.3.14.5)

The API call returns possible exceptions, warnings, errors, and a list of identified equipment entities with their sublots.

Specific API Support for Interfaces (SR1089.9+)

- ➢ For recent changes, see revision history (page 89).

> The following dedicated APIs are available to support interface, e.g. with a DCS adapter.
>
> For details related to DCS requests, see "Technical Manual DCS Adapter" [A13] (page 87).

Listener for Consumed Material Requests (SR1089.9.1)

- ➢ For recent changes, see revision history (page 89).

Consumed material requests can either request to process material consumptions (via DCS integration or phase-specific exception) or to remove a mistaken data set (only via phase-specific exception).

Process material consumption:

1. Check if the data received with the request points to exactly one order step input. Otherwise the request cannot be processed.
   If there are two or more order steps in a XOR branch containing the same material position, at least one of the order steps must be started before the system can process the request for an order step.
2. Identify the corresponding sublot for the order step input.
   If the sublot is not known, create a temporary sublot and continue with this sublot.
3. Set the request-specific UDAs at the order step input.
4. Update the sublot's quantity (consume quantity).

5. If applicable, delete the sublot logically.
6. Release the identified sublot.

Reverse an already processed consumption (only via phase-specific exception):

1. Check if the request is coming from an MES (e.g. phase building block).
   Otherwise the request cannot be processed.
2. Check if a sublot ID is specified in the request.
   Otherwise the request cannot be processed.
3. Check if the data received with the request points to one or more order step inputs.
   Otherwise the request cannot be processed.
4. Set the consumed quantity of the first matching order step input to 0.
5. Only if the corresponding sublot is neither logically deleted nor temporary and not identified by another order:
   Update its quantity.

Listener for Produced Material Requests (SR1089.9.2)

➢ For recent changes, see revision history (page 89).

Produced material requests contain information about the produced material (via DCS integration or via phase-specific exception).

Process produced material request

1. Before any produced material request can be processed, at least one sublot of the ingoing material (order step input) must have already been consumed.
2. Check if the data received with the request points to an exactly one order step output.
   Otherwise the request cannot be processed.
   If there are two or more order steps in a XOR branch containing the same material position, at least one of the order steps must be started before the system can process the request for an order step.
3. Load the sublot for the order step output.
   1. If the sublot is available, correct the quantity of the sublot and update the sublot’s transaction history.
   2. If the sublot is not known, create a new sublot.
      - Check if the order step runs.
        If so, assign the sublot to the storage location of the order step's work center.
        Otherwise assign the sublot to the default storage location.

- Check if the order step output is the final one.
  If so, use the batch ID of the order.
  Otherwise use the batch of the produced sublots. If there are no produced sublots, load/create a batch according to the request or create a new batch with a system-generated ID.
- Check if a sublot ID is specified in the request.
  If so, use the sublot ID.
  Otherwise sue a system-generated ID.
- Update the sublot's transaction history.

Listener for Batch Information Requests (SR1089.9.3)

➢ For recent changes, see revision history (page 89).

Batch information requests contain a batch identifier and its material to get all relevant batch information in the reply.

Process batch information request

- Check if the batch identifier received with the request matches an available batch. If not, the system includes the material identifier received with the request in its check for an available batch.
- If case of success, the system returns the batch identifier, material identifier, batch status, batch quantity, batch potency, production date, expiry date, and retest date of the batch.
- If no batch can be found, the system returns an error.

Signature for Phase Completion (SR1089.4+)

Signature panel (SR1089.4.1)

During execution, the system extends the phase by a phase completion signature panel, according to the phase-specific configuration during recipe and workflow authoring.
A second type of phase completion signatures are "pre-defined phase completion signatures". They are assigned to a phase and depend on the current context (e.g. **Quantity entry** weighing method). In this case, any other phase completion signature that has been assigned to the phase during recipe and workflow authoring is ignored.

Authorization (SR1089.4.2)

In case a phase completion signature is defined for a phase, the system requires an authorization by the appropriate users in order to complete a phase.
For a sequential phase completion signature, the system checks that the user has not signed for a previous phase completion within the operation. Otherwise, the system

creates a system-triggered exception. After signing the exception the user can sign for the phase completion.

Dynamic behavior (SR1089.4.3)

The system allows a phase to dynamically ignore the assigned signature during execution. Example use cases are: automated vs. manual tare, skipped phases.

Password performance (SR1089.4.4)

In case passwords have already been entered, but the operator returns to the phase rather than completing the phase right away, the system deletes the passwords and requires them to be entered again.

EBR User Experience (SR1089.5+)

Focus of input fields (SR1089.5.1)

During EBR execution, input fields that hold the focus are highlighted.

Monitoring and Recovery Capabilities (SR1200+)

- For recent changes, see revision history (page 89).

Access Control (SR1200.1.2)

The system allows to control the access to the PharmaSuite server monitoring tool.

Operation Execution (OE) Server (SR1200.1+)

Monitoring (SR1200.1.1)

The PharmaSuite server monitoring tool allows to monitor the status of an OE server including its current runtime operations.

Resuming Server-run Operations (SR1200.1.3)

Running operations can be held and resumed. Changing the status of running operations is controlled by one signature class.
Per default configuration, the change action must be confirmed by a single signature (see **Electronic Signatures (SR1095.50.2)** in "Functional Requirement Specification Non-functional Requirements" [A10] (page 87)).

After the electronic signature has been performed successfully, the system automatically records a "hold operation"-specific exception for the operation or adds a "resume operation"-specific comment to the recorded exception. The risk level of the exception can be configured.

For details, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Electronic Batch Recording (EBR) Server (SR1200.2+)

- For recent changes, see revision history (page 89).

Monitoring (SR1200.2.1)

- For recent changes, see revision history (page 89).

The PharmaSuite server monitoring tool allows to monitor the status of an EBR server including its current runtime procedures.

Unload and Reload Procedures of Orders and Workflows (SR1200.2.2)

- For recent changes, see revision history (page 89).

Procedures of orders and workflows running on the EBR server can be unloaded and reloaded later without shutting down the EBR server. Changing the loading-related status of running procedures is controlled by one signature class.
Per default configuration, the change action must be confirmed by a single signature (see **Electronic Signatures (SR1095.50.2)** in "Functional Requirement Specification Non-functional Requirements" [A10] (page 87)).

After the electronic signature has been performed successfully, the system automatically records an "unload procedure"-specific exception for all unit procedures of the order and workflow or adds a "load procedure"-specific comment to the recorded exception. The risk level of the exception can be configured.

For details, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Recovery Capabilities (SR1200.3+)

- For recent changes, see revision history (page 89).

PharmaSuite provides certain recovery capabilities to resolve issues during execution that are caused by a faulty or undefined recipe parameter setup (e.g. information flow expressions of a process parameter or transition conditions cannot be calculated by the system).

The recovery can be achieved by aborting a phase (in the Production Execution Client), repairing a phase by overriding parameter data (in the Production Execution Client), or forcing a specific transition (in the Production Management Client). The usage of each of these measures requires specific privileges and is tracked as an exceptional situation in the batch report.

For details related to forcing a transition, see **Force Execution Transition (SR1084.43)** for batch-specific orders and **Force Execution Transition (SR1085.2.13)** for workflows in "Functional Requirement Specification Runtime Data Management" [A4] (page 87).

Abort Phase (SR1200.3.1)

- For recent changes, see revision history (page 89).

The completion of an active phase can be triggered by tapping the **Abort** button in the Action column of the **Navigator (SR3071.4.2.3)** view (page 9). This results in a specific "Phase execution aborted" exception with which the system tracks the forced phase completion and adds it to the batch report.

- The exception text reads **"Phase execution aborted"** (configured in the **pec_ExceptionMessage** message pack (**PhaseExecutionSkippedExc_Msg**)).
- The default risk assessment of the "Phase execution aborted" exception is **high** (configured with the **PhaseExecutionSkippedExcRisk** configuration key).

  For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

- The signature privilege of the "Phase execution aborted" exception is defined with the **PEC_Signature_PhaseExecutionSkipped** access privilege.
- Merely aborting a phase bypasses all UI extensions (e.g. phase completion signature) and other phase-specific checks that may veto the completion.
- After the phase has been merely aborted, processing of the operation continues normally, according to the SFC modeling in the underlying recipe.

  The information flow will be broken if any output variable of a merely aborted phase is used later on in the control recipe. A merely aborted phase already represents its own instance of the phase, however, any output variable of a merely aborted phase is undefined except for the output variables provided by the framework (see **Output Variables (SR0900.9+)** requirement (page 84).

Repair Phase (SR1200.3.2)

➢ For recent changes, see revision history (page 89).

Repairing an active phase by overriding its process parameters can be triggered by tapping the **Repair** button in the Action column of the **Navigator (SR3071.4.2.3)** view (page 9).

- Repairing is not available for phases that are explicitly excluded from being repaired. The affected phases are maintained with the **PhaseRepairModeBlackList** configuration key.

  For configuration details, see chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A11] (page 87)

- Repairing is only available as long as no exceptions have been recorded for a phase. This restriction does not include "phase repair mode"-specific exceptions.

  Customer-specific implementations can exclude specific exceptions from this rule.

Repairing consists of the following steps:

1. The current phase instance is aborted automatically and the system records a specific "Phase repair mode started" exception that is assigned to the aborted instance of the phase and added to the batch report.

2. The system automatically creates a new instance of the phase that is first displayed in the repair mode that allows to override its process parameters.

   For editing the process parameter values, the repair mode provides the editors that are available in the Parameter Panel of Recipe and Workflow Designer, except for the Expression Editor. For this reason, e.g. an equipment object-related parameter cannot be repaired.

3. Upon completion of the repair mode, the system records a specific "Phase parameters repaired" exception that is assigned to the new instance of the phase and added to the batch report.
4. Finally, the new instance of the phase is displayed again with its regular user interface and can be processed as usual.

Starting the phase repair mode:

- The exception text reads **"Phase repair mode started"** (configured in the **pec_ExceptionMessage** message pack (**PhaseRepairModeStartedExc_Msg**)).
- The default risk assessment of the "Phase repair mode started" exception is **high** (configured with the **PhaseRepairModeExcRisk** configuration key).

  For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

- The signature privilege of the "Phase repair mode started" exception is defined with the **PEC_Signature_PhaseRepairMode** access privilege.
- For aborted phases, all UI extensions (e.g. phase completion signature) and other phase-specific checks that may veto the completion are bypassed.

Display of the new phase instance in the repair mode in the **Execution Window (SR3071.4.2.4.2)** view (page 12):

1. Repairing: <Phase name> [instance count]
   (indicated by a different text color)
2. Table of parameters
3. **Confirm** button

Completion of the phase repair mode:

- The exception text reads **"Phase parameters repaired"** (configured in the **pec_ExceptionMessage** message pack (**PhaseRepairModeCompletedExc_Msg**)).
  - For each modified parameter, the exception text is appended by:
    Parameter: <parameter identifier>
    Attribute: <attribute identifier>, original: <value>, new: <value>
  - If no value is defined, "N/A" is displayed.

  - If no parameter was modified at all, the exception text is appended by:
    No parameters changed.

- The default risk assessment of the "Phase parameters repaired" exception is **high** (configured with the **PhaseRepairModeExcRisk** configuration key).

> For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

- The signature privilege of the "Phase parameters repaired" exception is defined with the **PEC_Signature_PhaseRepairMode** access privilege.

Chapter 5

Automation Layer Integration (SR1086+)

Scales (SR1091.9)

The system shall be able to communicate with scales.

Scale Interface (SR1091.9.1)

It shall be possible to connect scales by a serial port interface.
At least four serial port interfaces shall be addressable by one dispensing work center.

Multi-range Scales (SR1091.9.2)

The system shall support multi-range scales. The following characteristics apply:

- Up to three ranges for each scale shall be supported.
- Each range shall have a specific precision.
- The ranges shall not be overlapping.
- The ranges do not have gaps between each other.

Supported Protocols (SR1091.9.3)

At least the following protocols/scales shall be supported:

- xBPI protocol of Sartorius
- SICS protocol of Mettler
- For details, see **Scale Drivers (SR1086.1)** details (page 64).

The system shall allow a system integrator to add support for additional protocols to an installed system.

It must be possible to add support for additional protocols with a minimized impact on the revalidation of the system.

It shall be possible to test additional protocol implementations and to validate the connectivity to the scales without using Dispense-specific phases.

Scale Drivers (SR1086.1)

The system shall be able to interface with scales of various vendors using their specific scale protocols. The following essential functions are mandatory:

- Get actual weight continuously
- Get last stable weight
- Zero scale

The following functions are important but optional (depending on the scale model):

- Tare scale
- Clear tare
- Manual tare
- Reset scale
- Disable scale value
- Read serial number
- Set nominal weight and tolerance
- Select scale connected to a terminal
- Start automatic internal calibration
- Start automatic external calibration
- Set dosage parameters

xBPI Protocol (SR1086.1.5)

The system shall be able to interface with Sartorius scales / indicators using the xBPI protocol.

SICS Protocol (SR1086.1.6)

The system shall be able to interface with Mettler Toledo scales using SICS protocol (incl. IND terminals from IND780 on).

Mettler Toledo ID1, ID2, ID3 Terminals (SR1086.1.7)

The system shall provide an ID1 driver that allows to interface with Mettler Toledo scale terminals ID1 [Plus/Minus], ID2, or ID3 [s] (MMR protocol-based).

Mettler Toledo ID5, ID7, IND Terminals (SR1086.1.11)

The system shall be able to interface with Mettler Toledo scale terminals ID5, ID7, or IND (including IND terminals up to IND690) (MMR protocol-based).

Mettler Toledo GD12 Scale Interface (SR1086.1.14)

The system shall be able to interface with Mettler Toledo scale interfaces GD12 (IDNet protocol-based).

MP8 Protocol (SR1086.1.15)

The system shall be able to interface with the Sartorius scales using MP8 protocol.

Ethernet Protocol (SR1086.1.16)

The system shall be able to interface with scales that are connected via Ethernet network (TCP/IP).

SBI4BP Protocol (SR1086.1.17)

The system shall be able to interface with Sartorius scales using SBI4BP protocol.

SBI4LA Protocol (SR1086.1.18)

The system shall be able to interface with Sartorius scales using SBI4LA protocol.

SBI4QC Protocol (SR1086.1.19)

The system shall be able to interface with Sartorius scales using SBI4QC protocol.

SBI4LP Protocol (SR1086.1.20)

The system shall be able to interface with Sartorius scales using SBI4LP protocol.

SBI4AC Protocol (SR1086.1.21)

The system shall be able to interface with Sartorius scales using SBI4AC protocol.

Label Printer (SR1086.3)

The system shall interface with label printers via standard MS Windows printer drivers.

Automation Integration (SR1086.4)

The system shall provide an Automation Integration framework that allows to read and write equipment-specific tags during the execution of control recipes or workflows, e.g. via Live Data / OPC server.

Supported Live Data Data Types (SR1086.4.1)

The system shall support the following Live Data data types (mapped to common OPC names):

- Integer (SHORT, LONG, CHAR, BYTE, USHORT, ULONG)
- Float (REAL4)
- Double (REAL8)
- String (STRING)
- Boolean (BOOLEAN)

The following table provides a more detailed view of the mapping:

FTLD ConstName= OPC standard data type	Description	Common OPC names	Mapped to Live Data data types	Mapped to PharmaSuite technical property
VT_EMPTY	Default/Empty (nothing)	---	N/A	N/A
VT_NULL	---	---	N/A	N/A
VT_I2	2-byte signed integer	**SHORT**	**Integer**	BigDecimal
VT_I4	4-byte signed integer	**LONG**	**Integer**	BigDecimal
VT_R4	4-byte (single precision) real	**REAL4**	**Float**	BigDecimal
VT_R8	8-byte (double-precision) real	**REAL8**	**Double**	BigDecimal
VT_CY	Currency	CURRENCY	not supported	N/A
VT_DATE	Date	DATE	not supported	N/A
VT_BSTR	Text (UNICODE)	**STRING**	**String**	String
VT_ERROR	Error code	---	N/A	N/A
VT_BOOL	Boolean (TRUE=-1, FALSE=0)	**BOOLEAN**	**Boolean**	Boolean
VT_I1	1-byte signed integer	**CHAR**	**Integer**	BigDecimal
VT_UI1	1-byte unsigned integer	**BYTE**	**Integer**	BigDecimal
VT_UI1	2-byte unsigned integer	**USHORT**	**Integer**	BigDecimal
VT_UI1	4-byte unsigned integer	**ULONG**	**Integer**	BigDecimal
VT_ARRAY	Arrays	ARRAY OF ...	not supported	N/A

Exception Management (SR1079+)

➢ For recent changes, see revision history (page 89).

Review by exception is one of the key features of an EBR system that allows to reduce time to market significantly. Review by exception, as a basis, requires a proper exception management handling on the shop floor during processing. In addition, the system has to offer capabilities for in-time reviews of exceptions on the shop floor. If specific exceptions should be documented and reviewed with an external QMS, the exception data can be transferred at the time of the exception creation.

This section provides the requirements relevant to exception management:

- **Exception categories (SR1079.1+)** requirements (page 67)
- **Exception recording (SR1079.2+)** requirements (page 73)

Exceptions need to be recorded to document irregular circumstances that have occurred during processing. They are an important element of the processing report (batch, device, workflow). Exception recording can be triggered either by a user, a phase, an operation, a unit procedure, or the system (e.g. Production Management Client, Operation Execution server). Independent of the category, each exception is linked to a phase, operation, or unit procedure. In the Exception Window, the system lists the exceptions and their comments. A header bar and a control bar complete the Exception Window.
Server-run-specific exceptions are only listed in the Details panel of the exception dashboard. See **Exception Dashboard (SR3200.3+)** in "Functional Requirement Specification Review and Approval" [A1] (page 87).

Exception Categories (SR1079.1+)

The following exception categories are available during exception recording. Additionally, specific behavior applies to some exceptional situations.

User-defined Exceptions (SR1079.1.1)

PharmaSuite supports free-text entries to be recorded as a user-defined exception of a phase or unit procedure. The following attributes are available:

- exception description,
- risk assessment,
- status, and
- **signature (1079.2.1)** data (page 74).

The available risk levels depend on the system configuration.

Exceptions of this category can be recorded for

- active phases via work area of the **Execution Window (SR3071.4.2.4.2)** view (page 12),
- completed phases via action column of the **Navigator (SR3071.4.2.3)** view (page 9),
- active unit procedures via the unit procedure-related **Actions (SR1089.8.1)** dialog (page 31).

Risk of the exception to patient safety, product quality, or data integrity: **High**, **Medium**, **Low**, **None**.
The availability of the **None** risk level depends on the system configuration.

The initial status is **Open**.

User-defined Exceptions in Production Response Client (SR1079.1.1.1)

For user-defined exceptions added in the Production Execution Client, the following additional attributes are available:

- result (mandatory to close an exception),
- CAPA ID: to reference corrective and preventive actions related to the exception, and
- reference to a specific phase, operation, or unit procedure for which the exception is recorded (mandatory).

The characteristics of following additional attributes have changed:

- status (mandatory) and
- risk assessment (mandatory).

Result: **---**, **Non-event**, **Note to record**, **Deviation**.

Status: **Open**, **In review**, **To be closed**, **Closed**. The initial status is **Open**.

Risk of the exception to patient safety, product quality, or data integrity: **High**, **Medium**, **Low**, **None**.

See also **Production Response Client (SR3071.5+)** and **Exception Dashboard (SR3200.3+)** in "Functional Requirement Specification Review and Approval" [A1] (page 87).

User-triggered Exceptions (SR1079.1.2)

PharmaSuite allows to manually trigger and record an exception that is defined by the respective phase as a user-triggered exception of a phase. The phase may provide several pre-defined alternative exceptions that cover expected exception cases (e.g. manual identification for active phases). The following attributes are available:

- exception description,
- risk assessment,
- status, and
- **signature (SR1079.2.1)** data (page 74).

These exceptions can be recorded for an active phase via work area of the **Execution Window (SR3071.4.2.4.2)** view (page 12).

The description and the risk that is attributed to the exception (**High**, **Medium**, **Low**, **None**) are controlled by the phase and can be configured by the process parameters in Recipe and Workflow Designer.

The initial status is **Open**.

Explicit capability-related (SR1079.1.2.1) exceptions (page 69) are also user-triggered exceptions.

Explicit Capability-related Exceptions (SR1079.1.2.1)

Depending on certain capabilities that are configurable in Recipe and Workflow Designer or in the system configuration, explicit capability-related exceptions can relate to phases, operations, or unit procedures. They are triggered by the following Cockpit actions:

- Detach with **detach-specific (SR1089.6.1.2)** exceptions (page 35) for operations,
- Actions for event-triggered operations with **ETO-specific (SR1089.6.2.9+)** exceptions (page 38),
- Pause unit procedure with **pause unit procedure-specific (SR1089.8.3.2)** exception (page 33),
- Detach with **detach-specific (SR1089.8.4.2)** exceptions (page 31) for unit procedures,
- Abort and reactivate unit procedure with **reactivate unit procedure-specific (SR1089.8.5.1)** exception (page 34), and
- Append workflow to a unit procedure with append workflow-specific exception **during unit procedure processing** (SR1089.8.2.2) (page 33) or **at workflow start** (SR1089.7.1.2) (page 30).

The recording during execution follows the same UI pattern as the **user-triggered exceptions (SR1079.1.2)** of phases (page 69).

Post-completion Exceptions (SR1079.1.3)

PharmaSuite allows to manually trigger and record an exception that is defined by the respective phase as a post-completion exception of a phase. The phase may provide several pre-defined alternative exceptions that cover expected exception cases (e.g. label reprint for completed phases). The following attributes are available:

- exception description,
- risk assessment,
- status, and
- **signature (1079.2.1)** data (page 74).

These exceptions can be recorded for a completed phase via action column of the **Navigator (SR3071.4.2.3)** view (page 9).

> The description and the risk that is attributed to the exception (**High**, **Medium**, **Low**, **None**) are controlled by the phase and can be configured by the process parameters in Recipe and Workflow Designer.
>
> The initial status is **Open**.

Locking of Operations (SR1079.1.3.1)

In case a post-completion exception is triggered for an already completed operation, the system locks the related operation until the recording of the exception has been completed.

System-triggered Exceptions (SR1079.1.4)

PharmaSuite supports exceptions that are automatically triggered by the phase based on the data recorded during execution as a system-triggered exception of a phase. The phase may provide several pre-configured or on-phase-level-configured exceptions that cover expected exception cases (e.g. limit violation). In an **exception dialog (SR1089.3.8.1)** UI element (page 45) with a warning or error message, the phase prompts the user to record the exception. The following attributes are available:

- exception description,
- risk assessment,
- status, and
- **signature (1079.2.1)** data (page 74).

These exceptions can be recorded for an active phase via work area of the **Execution Window (SR3071.4.2.4.2)** view (page 12).

The description and the risk that is attributed to the exception (**High**, **Medium**, **Low**, **None**) are controlled by the phase and can be configured by the process parameters in Recipe and Workflow Designer.

The initial status is **Open**.

Implicit capability-related (SR1079.1.4.2) exceptions (page 72) are also system-triggered exceptions.

The execution of the following use cases in the Production Management Client or Data Manager results in an automatic recording of a system-triggered exception. The signatures applied for the confirmation of the use case are also used to record the exceptions:

- Unlock and unregister an operation with the **Remove Object Locks** task, see **Unlock S88 Runtime Operation (SR3071.3.6.1)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87).
- Change the list of workflows appended to an order step with the **Manage Orders** use case, see **Append Workflows - Related Exceptions (SR1084.34.1)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87).
- Reactivate non-Dispense order step with the **Manage Orders** use case, see **Reactivate Non-Dispense Order Step - Related Exceptions (SR1084.35.1)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87).
- Abort non-Dispense order step with the **Manage Orders** use case, see **Abort Non-Dispense Order Step - Related Exceptions (SR1084.36.1)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87).
- Cancel order/workflow or abort unit procedure with the **Manage Orders/Workflow** use cases, see **Cancel Order (SR1084.3)**, **Abort Non-Dispense Order Step - Related Exceptions (SR1084.36.1)**, **Cancel Order (SR1084.100.7)**, **Abort Device Order Step - Related Exceptions (SR1084.101.4.1)**, and **Cancel Workflow (SR1085.2.1)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87).
- Finish order with aborted order steps with the **Manage Orders** use case, see **Finish Order With Aborted Order Steps - Related Exceptions (SR1084.42.1)** in "Functional Requirement Specification Runtime Data Management" [A4] (page 87).
- Force unbinding of an equipment entity in Data Manager, see **Force unbinding an equipment entity (SR1099.3.3.7)** in "Functional Requirement Specification Data Management" [A7] (page 87).
- Save changes of runtime data, such as runtime properties, graph-related data (e.g. status, expiry date), and grouping situations of a bound and read-only equipment entity in Data Manager, see **Change Properties of Equipment Entity in Read-only Status (SR1099.3.3.8)** in "Functional Requirement Specification Data Management" [A7] (page 87).

Exceptional Actions (SR1079.1.4.1)

System-triggered exceptions support use cases that require certain system actions by a phase when the exception has been signed.

> Example: Perform and record a sublot identification after the failed quality status check has been signed as an exception.

Implicit Capability-Related Exceptions (SR1079.1.4.2)

➢ For recent changes, see revision history (page 89).

Dependent on certain capabilities that are configurable in Recipe and Workflow Designer, implicit capability-related exceptions can relate to operations. They are triggered automatically by the system with

- **Escalation-specific (SR1089.6.3.4)** exceptions (page 39),
- **Operation Execution (OE) server-run-specific (SR1089.6.4.2)** exceptions (page 40), and
- **Triggered Operation Management (TOM) server-run-specific (SR1089.6.4.3)** exceptions (page 40).

The recording during execution is done without user interaction.

System-triggered Error Messages (SR1079.1.6)

PharmaSuite supports error messages that are automatically displayed by the phase based on the data recorded during execution (e.g. fixed checks during material identification). The display of the messages is configured on phase level.

The error message dialog does not provide a button that allows to record any exception.

System-triggered versus User-triggered (SR1079.1.7)

In case a system-triggered exception is being recorded, the Exception Window does not show the optional user-triggered exceptions.

Only the list of already existing exceptions is shown (if applicable) along with the exception panel that allows to sign the actual exception to be recorded.

One User-triggered Exception at a Time (SR1079.1.8)

After a user-triggered exception has been selected for recording, all other available user-triggered exceptions displayed in the Exception Window are disabled.

They are enabled again once the Exception Window is closed (with the **Back** button) and reopened (with the **Exception** button).

Default Exception Text (SR1079.1.9)

The exception description of any exception is a concatenation of

- the exception text defined with the related process parameter and
- the exception-specific information added by a phase (e.g. old/new values).

In case no exception text has been configured in the recipe as part of the related process parameter, the system adds a **Default exception text.** during execution.

Exception Recording (SR1079.2+)

The following features are relevant to exception recording:

- **Cancelation of exception (SR1079.2.7)** feature (page 73)
- **Signatures for exceptions and comments (SR1079.2.1)** feature (page 74)
- **List of exceptions (SR1079.2.2)** feature (page 75)
- **Comments related to an exception (SR1079.2.3)** feature (page 76)
- **Bookmark (SR1079.2.4)** feature (page 77)
- **Exception marker (SR1079.2.5)** feature (page 77)
- **Exception - Page break (SR1079.2.6)** feature (page 78)
- **Inform an External QMS about New Exceptions (SR1079.2.8)** feature (page 78)
- **Exception Updated by an External QMS (SR1079.2.9)** feature (page 79)

Cancelation of Exception (SR1079.2.7)

➢ For recent changes, see revision history (page 89).

The system prevents that an operator cancels an exception without adding this information to the batch report. This applies to any phase-specific exception of the **user-triggered (SR1079.1.2)** category (page 69), **system-triggered (SR1079.1.4)** category (page 70), or **post-completion (SR1079.1.3)** category (page 70).
The cancelation of an exception can be triggered by tapping the **Back** button in the Exception Window. This results in the display of a question dialog. Upon confirmation, a specific "Exception canceled" exception needs to be signed with which the system tracks the cancelation of the original exception (e.g. related to a limit violation) and adds it to the batch report.

- The exception category remains the same as the category of the original exception that was canceled.

- The exception text of the original exception is kept, it is, however prepended with "**Exception canceled.**" (configured in the **wad_ExceptionMessage** message pack (**canceledException_Msg**).
- The default risk assessment of the "Exception canceled" exception is **low** (configured with the **CanceledExceptionRisk** configuration key).

  For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

- The signature privilege of the "Exception canceled" exception is defined with the **PEC_Signature_CanceledException** access privilege.

By default, there is no **Cancel** button available in the exception dialog of a system-triggered or user-triggered exception. For configuration details, see chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A11] (page 87).

Signatures for Exceptions and Comments (SR1079.2.1)

Depending on the system configuration, exceptions and their comments can be confirmed by a single or double signature (see **Electronic Signatures (SR1095.50.2)** in "Functional Requirement Specification Non-functional Requirements" [A10] (page 87)).

The Production Execution Client uses the reason(s) defined with the **Reason (1st)** and **Reason (2nd)** attributes of the signature privilege process input parameter of a master recipe or master workflow in the following cases:

- **Phase-completion (SR1089.4+)** signatures (page 55)
  This does not apply to pre-defined phase completion signatures.
- **System-triggered (SR1079.1.4)** exceptions (page 70)
- **User-triggered (SR1079.1.2)** exceptions (page 69)
- **Explicit capability-related (SR1079.1.2.1)** exceptions (page 69)

In case no reason has been defined in the master recipe or master workflow, the default value defined in the access privilege is used.

The following configuration capabilities are not applicable to signatures for exceptions and their comments:

- Comments (no signature-related comments allowed, see **Comments (SR1095.50.2.1.1)** in "Functional Requirement Specification Non-functional Requirements" [A10] (page 87))
- Signature level (no alternative warnings/errors, implicit or disabled signatures allowed, see **Configuring Electronic Signatures (SR1095.50.2.2)** in "Functional Requirement Specification Non-functional Requirements" [A10] (page 87))

Password Performance (SR1079.2.1.5)

In case passwords have already been entered, but the operator returns to any input field of the exception/comment panel, the system deletes the passwords and requires them to be entered again, before the exception/comment can be finally added.

Signatures for User-defined Exceptions (SR1079.2.1.1)

The signatures that are required to sign an exception of the **user-defined (SR1079.1.1)** category (page 67) are controlled by a different global signature privilege for both the Production Execution Client and Production Response Client.

Signatures for Phase-defined Exceptions (SR1079.2.1.3)

The signature that is required to sign a phase-defined exception of the **user-triggered (SR1079.1.2)** category (page 69), **system-triggered (SR1079.1.4)** category (page 70), or **post-completion (SR1079.1.3)** category (page 70) is controlled by the **Risk assessment** attribute of the exception-related process parameter of the phase.

In case no signature privilege is configured on phase level, the signature is controlled by a global default signature privilege per specific risk (**High**, **Medium**, **Low**, **None**).

Signatures for Explicit Capability-related Exceptions (SR1079.2.1.4)

The signature that is required to sign exceptions of the **explicit capability-related (SR1019.1.2.1)** type (page 69) is controlled by the **Risk assessment** attribute of the related capability and the matching signature privilege that is assigned to this specific procedural element.

> During recipe authoring in Recipe Designer, the system assures that a matching signature is available for each capability-related exception that has been enabled.

Signatures for Exception-related Comments (SR1079.2.1.2)

The signatures that are required to sign an exception-related comment are controlled by a different global signature privilege for both the Production Execution Client and Production Response Client.
This does not apply to **mandatory exception-related comments in the Production Execution Client** (SR1079.2.3.4) (page 76).

List of Exceptions (SR1079.2.2)

In the Exception Window of a phase, the system lists the exceptions and their comments related to this phase. This applies to active phases (accessible via **control bar (SR3071.4.2.4.3)** area (page 6)) and completed phases (accessible via **Navigator (SR3071.4.2.3)** view (page 9)).

The identifier of an exception is an ascending 3-digit number. It is generated by the system and unique within the context of a batch. If the number of exceptions exceeds the range, the number of digits of the identifier is increased.

Unit Procedure-specific Exceptions (SR1079.2.2.1)

In the Exception Window of a unit procedure, the system lists the exceptions and their comments related to this unit procedure, its operations, and phases. The Exception Window is accessible as a **Context-related (SR1089.8.1)** action (page 31) via the unit procedure-specific context dialog. An operator can add comments to existing exceptions and can record new exceptions related to this unit procedure.

The creation of identifiers of exceptions and comments follows the same pattern as for phase-specific exceptions (**List of Exceptions (SR1079.2.2)** feature (page 75), **Comments related to an Exception (SR1079.2.3)** feature (page 76)).

Comments Related to an Exception (SR1079.2.3)

Comments can be added per exception. The following attributes are available:

- comment description,
- risk assessment (optional),
- status (optional), and
- **signature (1079.2.1)** data (page 74).

The editability of the risk level depends on the system configuration.

The identifier of a comment is a concatenation of the exception identifier, a hyphen (-), and an ascending 2-digit number.

Comments related to an exception can be added to exceptions of active phases, completed phases, and idle or active operations and unit procedures.

Status: **Open**, **In review**, **To be closed**, **Closed** (only available in the Production Response Client).

Risk: **None**, **Low**, **Medium**, **High**.
The availability of the **None** risk level depends on the system configuration.

Mandatory Comments in Production Execution Client (SR1079.2.3.4)

Within the Production Execution Client, in case an exception-related comment is defined as mandatory in the recipe/workflow, both the exception and the comment are recorded with a single signature. The system does not allow to record the exception without a comment.

Attribute Propagation in Production Execution Client (SR1079.2.3.1)

Within the Production Execution Client, the following attributes of a comment are propagated to the related exception:

- risk assessment and
- status (**Closed** is not available).

The editability of the risk level depends on the system configuration.

> Risk: **None**, **Low**, **Medium**, **High**.
> The availability of the **None** risk level depends on the system configuration.
>
> Status: **Open**, **In review**, **To be closed**.

Risk Tracking (SR1079.2.3.2)

In case the risk assessment of an exception is changed by adding a comment, the previous risk assessment is automatically rendered into the description of the comment.

Attribute Propagation in Production Response Client (SR1079.2.3.3)

Within the Production Response Client, the following attributes of a comment are propagated to the related exception:

- risk assessment,
- status,
- result, and
- CAPA ID.

> Risk: **None**, **Low**, **Medium**, **High**.
>
> Status: **Open**, **In review**, **To be closed**, **Closed**.
>
> Result: **---**, **Non-event**, **Note to record**, **Deviation**.
>
> See also **Details Panel (SR3200.3.3)** in "Functional Requirement Specification Review and Approval" [A1] (page 87).

Bookmark (SR1079.2.4)

As long as comments can be added to an exception, the exception can also be bookmarked to simplify the review process.

Exception Marker (SR1079.2.5)

Phases for which an exception has been recorded are displayed with a red upper left corner as exception marker in the work area of the **Execution Window (SR3071.4.2.4.2)** view (page 12) and in the **Navigator (SR3071.4.2.3)** view (page 9). This applies to active and completed phases.

In phases that display values in a table, cells of values for which an exception has been recorded can be displayed with a red upper left corner as exception marker.

Exception - Page Break (SR1079.2.6)

In case the exception text exceeds 2048 characters, the system automatically creates new exception-related comments that hold the remaining text (up to another 2048 characters per comment).

Inform an External QMS about New Exceptions (SR1079.2.8)

➢ For recent changes, see revision history (page 89).

In the default configuration, the feature to inform an external QMS is disabled.

In case the feature is enabled, the system shall send an event after the successful creation of an exception. The default rule to trigger an event depends on the risk assessment of the exception. The system allows to configure the risk levels for which an event is sent.

> For details related to the risk level, see chapter "Defining the Risk Level for Exceptions" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

When an event for an exception is sent, the status of the exception is set to **In review** instead of **Open** and the communication status is set to **Sent**.

The event contains the following data:

- Unique exception ID (by default, workflow or order identifier concatenated by ' - ' and the exception identifier). The ID is stored in the exception, since the **Exception updated by an external QMS (SR1079.2.9)** feature (page 79) uses the ID to determine the exception.
- Workflow or order identifier
- Target batch identifier (empty in case of workflows)
- Target batch material number (empty in case of workflows)
- Exception identifier
- Exception comment (if an exception comment is split due to its length, the event contains the complete comment)
- Exception reference
- Exception risk
- Exception category
- First signer of exception (date, first name, last name, login name, email)

- Second signer of exception (date, first name, last name, login name, email)
- Station on which sending the exception was triggered. Can be empty.
- Work center on which the linked object was executed during exception creation

The data of the event can be extended.

> For details, see chapter "Supporting External Exception Review" in Volume 2 of the "Technical Manual Configuration and Extension" [A5] (page 87).

Exception Updated by an External QMS (SR1079.2.9)

➢ For recent changes, see revision history (page 89).

The system allows an external QMS to update exceptions as long as the related order or workflow is not in the **Reviewed**, **Canceled**, or **Annulled** statuses. The exception status and the communication status can be updated and an external ID can be set. The update is added as a comment to the exception with an implicit signature of the default system user. The description of the comment with the information that it was created by the external QMS can be expanded by a text with a maximum length of 1,500 characters. In case the text is longer, it will be truncated (indicated by …).

Additionally, the exception dashboard is updated.

Chapter 7

Phase-related Common Capabilities (SR0900+)

This chapter describes specific requirements that are common to all PharmaSuite phase building blocks.
Its structure matches the common structure of each Functional Requirement Specification of a phase building block.

Phases allow an operator to perform specific tasks during execution.

Example use cases are:

- Identification of material for dispensing.
- Documenting process-related data.
- Displaying specific instructions.

For details, please refer to the phase-specific Functional Requirement Specifications.

Layout

The phase provides individual layouts for its representation during execution, in the Navigator (page 82), and in the sub-report (page 83).

Representation during Execution (SR0900.1+)

➢ For recent changes, see revision history (page 89).

The representation during execution depends on the phase mode.

Preview mode

Provided by phase-specific implementation.

Active mode

Provided by phase-specific implementation.

Completed mode

Provided by phase-specific implementation.

Completed mode after recovery (SR0900.1.1)

➢ For recent changes, see revision history (page 89).

Applies only if the phase-specific implementation of the **Completed** mode could not be rendered for technical reasons, since the phase was merely aborted or aborted for repairing.

- <Phase name> [instance count]: The phase was aborted.
  If the phase has been merely aborted with **recovery (SR1200.3+)** capabilities (page 58).
- <Phase name> [instance count]: The phase was aborted for repairing.
  If the phase has been aborted for repairing with **recovery (SR1200.3+)** capabilities (page 58).

Representation in Navigator (SR0900.4+)

The Navigator provides the following details:

Phase column (SR0900.4.1)

- <Phase name>
  - Example:
    Read processing value

Information column

Provided by phase-specific implementation.

Information column after recovery (SR0900.4.3)

➢ For recent changes, see revision history (page 89).

- Aborted
  If the phase has been merely aborted or aborted for repairing with **recovery (SR1200.3+)** capabilities (page 58).

Action column

➢ For recent changes, see revision history (page 89).

Provided by phase-specific implementation (applies to completed phases only).

Action column of Active phases (SR0900.4.2)

➢ For recent changes, see revision history (page 89).

- Abort, provides access to **recovery (SR1200.3+)** capabilities (page 58).

- Repair, provides access to **recovery (SR1200.3+)** capabilities (page 58).
  For the availability of the **Repair** button, see **Repair Phase (SR1200.3.2)** capability (page 59).

Action column after recovery (SR0900.4.3)

➢ For recent changes, see revision history (page 89).

For phases that have been merely aborted or aborted for repairing with **recovery (SR1200.3+)** capabilities (page 58), none of the phase-specific actions are available.

Representation in Sub-report (SR0900.5+)

The sub-report contains the following information:

Common sub-report elements (SR0900.5.1)

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

Sub-report elements

Provided by phase-specific implementation.

Sub-report elements after recovery (SR0900.5.2)

➢ For recent changes, see revision history (page 89).

Applies only if the phase-specific implementation of the sub-report elements could not be rendered for technical reasons, since the phase was merely aborted or aborted for repairing.

- The phase was aborted.
  If the phase has been merely aborted with **recovery (SR1200.3+)** capabilities (page 58).
- The phase was aborted for repairing.
  If the phase has been aborted for repairing with **recovery (SR1200.3+)** capabilities (page 58).

Process Parameters (SR0900.8+)

The following process parameters define the behavior of the phase.

EDITOR-SPECIFIC PARAMETERS

List editor (SR0900.8.3)

The system provides a List editor for entering list items.

Option List editor (SR0900.8.2)

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

Property Selection editor (SR0900.8.1)

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

Trigger Selection editor (SR0900.8.4)

The system provides a Trigger Selection editor for selecting triggers based on their equipment graph. The graph purpose determines which triggers of which equipment graphs are available. System triggers and triggers of graphs in the **Archived** status are not available.

Output Variables (SR0900.9+)

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

Instance count (SR0900.9.1)

- 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 (SR0900.9.2)

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

Completion time (SR0900.9.3)

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

Identifier (SR0900.9.4)

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

Reference Documents

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

No.	Document Title	Part Number
A1	PharmaSuite Functional Requirement Specification Review and Approval	PSFRSRA-RM004E-EN-E
A2	PharmaSuite Functional Requirement Specification Recipe and Workflow Management	PSFRSRD-RM008E-EN-E
A3	PharmaSuite Functional Requirement Specification Production Execution	PSFRSPE-RM008E-EN-E
A4	PharmaSuite Functional Requirement Specification Runtime Data Management	PSFRSRT-RM004E-EN-E
A5	PharmaSuite Technical Manual Configuration & Extension - Volume 2	PSCEV2-GR008E-EN-E
A6	PharmaSuite Technical Manual Installation - Enterprise Edition	PSEN-IN008E-EN-E
A7	PharmaSuite Functional Requirement Specification Data Management	PSFRSDM-RM004E-EN-E
A8	PharmaSuite Functional Requirement Specification Dispense and Inline Weighing	PSFRSDI-RM006E-EN-E
A9	PharmaSuite Functional Requirement Specification Material Tracking Phases	PSFRSMT-RM005E-EN-E
A10	PharmaSuite Functional Requirement Specification Non-functional Requirements	PSFRSNF-RM001E-EN-E
A11	PharmaSuite Technical Manual Configuration & Extension - Volume 4	PSCEV4-GR008E-EN-E
A12	PharmaSuite Technical Manual Developing System Building Blocks	PSBB-PM007E-EN-E
A13	Technical Manual DCS Adapter	DCTMAD-GR001C-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
Functional Requirement Specification	1.1

Revision History

The following table describes the history of this document.

Changes related to the document:

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

Changes related to "Production Execution Client" (page 3):

Object	**Description**	**Document**
Cockpit (SR3071.4.1.1) (page 6)	Update The Cockpit also lists the unit procedures and operations of an unloaded process (order, workflow) even though they cannot be processed. PharmaSuite supports the concept of confidential objects to protect the intellectual property of recipes, workflows, orders, and related data from unauthorized access.	1.0
Server Heartbeat Check (SR3071.4.1.2) (page 16)	Update The EBR server also processes incoming messages from a Distributed Control System.	1.0
Navigator (SR3071.4.2.3) (page 9)	Update For active phases, the Action column provides access to **recovery (SR1200.3+)** capabilities.	1.0
User Access to Station Constraints (SR3071.4.8.3) (page 13)	Update PharmaSuite supports the concept of confidential objects to protect the intellectual property of recipes, workflows, orders, and related data from unauthorized access.	1.0

Changes related to "Electronic Batch and Workflow Recording" (page 21):

Object	**Description**	**Document**
Execution Framework - Basics (SR1089.1) (page 21)	Editorial For clarification: The system sets the order/workflow status to **Finished** when the last order/workflow step **of a recipe/workflow path** of an order/workflow is finished. No change of code.	1.0
System Recovery after Server Restart (SR1089.1.12) (page 28)	Update The EBR server also processes incoming messages from a Distributed Control System.	1.0
Create a New Run-specific Exceptions (SR1089.6.2.3.1) (page 37)	New The system automatically records a “create new run”-specific system-triggered exception when the Triggered Operation Management (TOM) server is not available when a new run is to be created. The risk level of the exception can be configured.	1.0
Remove a Template (SR1089.6.2.8) (page 37)	Update If the Operation Execution (OE) server, the Triggered Operation Management (TOM) server, or the message broker do not run, the template cannot be removed. The system displays an appropriate message.	1.0

Object	Description	Document
Operation Execution (OE) Server-run-specific Exceptions (SR1089.6.4.2) (page 40)	Update For clarification, requirement renamed and "The exception is added on operation level" added. No change of code.	1.0
Triggered Operation Management (TOM) Server-run-specific Exceptions (SR1089.6.4.2) (page 40)	New The system supports the recording of a server-run-specific exception in case a template that runs on a TOM server is restarted. The exception is added on unit procedure level.	1.0
Start and Append Workflows (SR1089.7.1) (page 29)	Update PharmaSuite supports the concept of confidential objects to protect the intellectual property of recipes, workflows, orders, and related data from unauthorized access.	1.0
Select Unit Procedure to Append Workflow (SR1089.7.1.1) (page 29)	Update PharmaSuite supports the concept of confidential objects to protect the intellectual property of recipes, workflows, orders, and related data from unauthorized access.	1.0
Appending Workflows to a Unit Procedure (SR1089.8.2) (page 31)	Update PharmaSuite supports the concept of confidential objects to protect the intellectual property of recipes, workflows, orders, and related data from unauthorized access.	1.0
Selecting Workflow to Append to Unit Procedure (SR1089.8.2.1) (page 32)	Update PharmaSuite supports the concept of confidential objects to protect the intellectual property of recipes, workflows, orders, and related data from unauthorized access.	1.0
Specific API Support for Interfaces (SR1089.9+) (page 53)	New PharmaSuite provides specific API support for interfaces.	1.0
Listener for Consumed Material Requests (SR1089.9.1) (page 53)	New PharmaSuite provides a listener specific for consumed material requests.	1.0
Listener for Produced Material Requests (SR1089.9.2) (page 54)	New PharmaSuite provides a listener specific for produced material requests.	1.0
Listener for Batch Information Requests (SR1089.9.3) (page 55)	New PharmaSuite provides a listener specific for batch information requests.	1.0

Changes related to "Monitoring and Recovery Capabilities" (page 57):

Object	Description	Document
Monitoring and Recovery Capabilities (SR1200+) (page 57)	Update Renamed since PharmaSuite also provides recovery capabilities within the Production Execution Client.	1.0
Electronic Batch Recording (EBR) Server (SR1200.2+) (page 57)	New The system provides monitoring capabilities for the EBR server.	1.0
Monitoring (SR1200.2.1) (page 57)	New The system provides monitoring capabilities for the EBR server.	1.0
Unload and Reload Procedures of Orders and Workflows (SR1200.2.2) (page 58)	New The system allows to unload and reload procedures of orders and workflows that are running on the EBR server.	1.0
Recovery Capabilities (SR1200.3+) (page 58)	New PharmaSuite provides recovery capabilities within the Production Execution Client.	1.0
Abort Phase (SR1200.3.1) (page 58)	New PharmaSuite provides recovery capabilities within the Production Execution Client.	1.0
Repair Phase (SR1200.3.2) (page 59)	New PharmaSuite provides recovery capabilities within the Production Execution Client.	1.0

Changes related to "Automation Layer Integration" (page 63):

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

Changes related to "Exception Management" (page 67):

Object	Description	Document
Exception Management (SR1079+) (page 67)	Context information-related New feature to transfer exceptions to an external QMS.	1.0
Implicit Capability-Related Exceptions (SR1079.1.4.2) (page 72)	Update Triggered Operation Management (TOM) server-run-specific exceptions are implicit capability-related exceptions.	1.0

Object	Description	Document
Cancelation of Exception (SR1079.2.7) (page 73)	New The system prevents that an operator can cancel the recording of an exception of the **user-triggered (SR1079.1.2)**, **system-triggered (SR1079.1.4)**, or **post-completion (SR1079.1.3)** category by creating an "Exception canceled" exception.	1.0
Inform an External QMS about New Exceptions (SR1079.2.8) (page 78)	New New feature to inform an external QMS about new exceptions.	1.0
Exception Updated by an External QMS (SR1079.2.9) (page 79)	New New feature to update exceptions through an external QMS.	1.0
Inform an External QMS about New Exceptions (SR1079.2.8) (page 78)	Update Data contained in the event: the bookmark data is not contained, the station refers to the station on which sending the exception was triggered. Can be empty. No code changed.	1.1
Exception Updated by an External QMS (SR1079.2.9) (page 79)	Update The description of the comment with the information that it was created by the external QMS can be expanded by a text with a maximum length of 1,500 characters. In case the text is longer, it will be truncated (indicated by ...). No code changed.	1.1

Changes related to "Phase-related Common Capabilities" (page 81):

Object	Description	Document
Representation during Execution (SR0900.1+) (page 81)	New Grouping requirement for the Representation during Execution phase-related common capability.	1.0
Completed Mode after Recovery (SR0900.1.1) (page 82)	New PharmaSuite provides recovery capabilities within the Production Execution Client.	1.0
Action Column (page 82)	Editorial The phase-specific implementation applies only to completed phases.	1.0
Action Column of Active Phases (SR0900.4.2) (page 82)	New For active phases, the system provides access to the **recovery (SR1200.3+)** capabilities.	1.0

Object	Description	Document
Information Column after Recovery (SR0900.4.3) (page 82)	New PharmaSuite provides recovery capabilities within the Production Execution Client.	1.0
Action Column after Recovery (SR0900.4.4) (page 83)	New PharmaSuite provides recovery capabilities within the Production Execution Client.	1.0
Sub-report Elements after Recovery (SR0900.5.2) (page 83)	New PharmaSuite provides recovery capabilities within the Production Execution Client.	1.0

Index