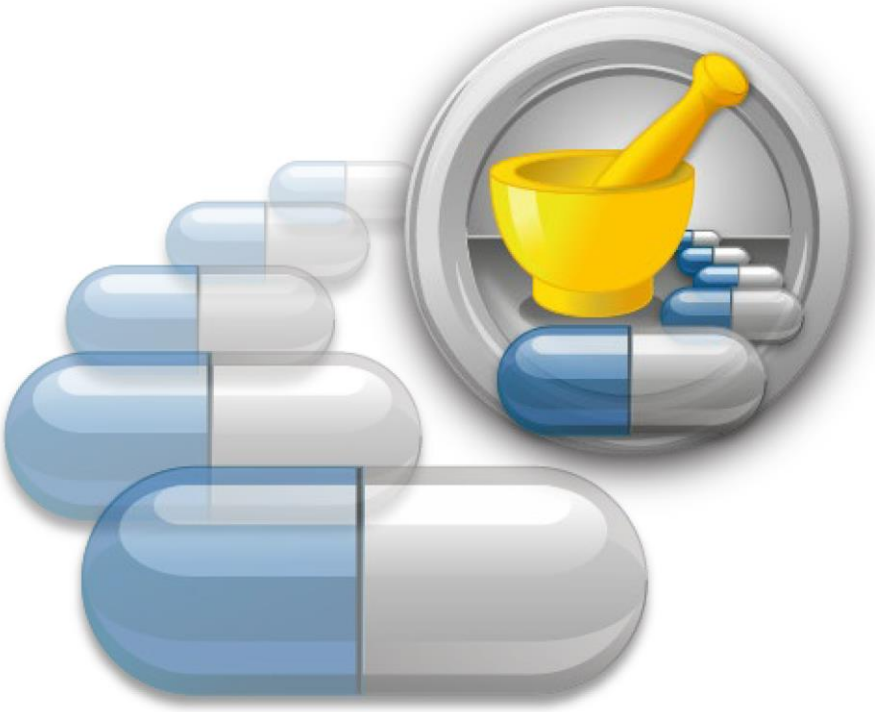
PharmaSuite®



RECIPE AND WORKFLOW MANAGEMENT

RELEASE 10.02.00

FUNCTIONAL REQUIREMENT SPECIFICATION

PUBLICATION PSFRSRD-RM010C-EN-E-SEPTEMBER-2021
Supersedes publication PSFRSRD-RM010B-EN-E

Rockwell Automation

Contact Rockwell See contact information provided in your maintenance contract.

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

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

Contents

Figures

Chapter 1

Introduction

This document details the requirements of the functions implemented related to recipe and workflow management with PharmaSuite Recipe and Workflow Designer.

Each requirement is composed of a name and a unique identifier (e.g., Procedural structure (SR3146.9.4)). If a requirement's meaning is for requirement grouping only, the identifier is appended by a plus sign (e.g., Messages (SR3146.9.2.16+)).

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 173) lists the changes made to the document with PharmaSuite 10.01.00 as the comparison baseline. Changes related to a requirement are marked as "Editorial", "Update", "New", or "Deleted", changes to the additional context information are marked as "Context information-related".

Typographical Conventions

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

Bold typeface Designates user interface texts, such as

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

Chapter 2

Recipe and Workflow Management (SR3146+)

In PharmaSuite, recipe and workflow management is done by means of the **Designer Workbench (SR3146.9+)** tool (page 129).

S95 defines a master recipe as follows: The master recipe is that level of recipe that is targeted to a process cell or a subset of the process cell equipment.
In its role as container for both the what (to manufacture) and the how (to manufacture), the master recipe is the place where the flow of the materials through the production process is specified. Thus, a master recipe typically contains all pharmaceutically-relevant data required to support GxP-compliant production. Master recipes are under version and status control.

Apart from master recipes PharmaSuite supports the definition and execution of workflows that also follow the S88 structure of a master recipe (master workflow).

A building block component can be an element created by the recipe author on the Procedure, Unit Procedure, Operation and Phase structure levels. These building blocks are custom building blocks and under full control of the recipe author. A building block element provided by the supplier with PharmaSuite on the Phase structure level is a system building block and is usable but not changeable by the recipe author. A system building block is always in an **Approved** status and has no revision.

Custom building blocks on Unit Procedure and Operation structure level can be allowed to be used as a **Procedural structure - Group of Building Blocks (SR3146.9.4.8)** elements (page 17).

Recipe and Workflow Designer of PharmaSuite is a graphical workbench for building and maintaining master recipes, master workflows, and their component building blocks. With its functions for status and version control, it covers the entire life cycle of a master recipe, master workflow, or custom building block. The workbench provides material flow control on the basis of material parameters, information flows for values from process parameters, privilege parameters for access rights, capability parameters, and equipment parameters.
For each structure level of a master recipe, master workflow, or building block, the system presents an SFC graph (sequential function chart).
PharmaSuite Recipe and Workflow Designer is available in two modes: as Recipe Designer and as Workflow Designer.

Most of the requirements that are related to Recipe Designer also apply to Workflow Designer. If a requirement applies only to a subset (Recipe Designer, Workflow Designer), this is explicitly noted within the requirement.

CONFIDENTIAL OBJECTS
The concept of confidential objects protects the intellectual property of recipes, workflows, custom building blocks, orders, ERP BOMs, and related data from unauthorized access. The system provides a specific access rights type to define access privileges that allow users to maintain protected recipes, workflows, and orders.

A master recipe or master workflow can be protected by assigning an explicit access privilege or inherit the access privilege from the used ERP BOM or the selected material

(product). Orders and workflows based on a protected master recipe or master workflow inherit the access privilege automatically. A user without a protected-specific access privilege is not able to

- view a protected master recipe or master workflow,
- start a protected order or workflow for execution
- log in at a station on which a protected order or workflow runs,
- see a protected order or workflow in the Production Response Client
- print the batch report, weighing report, or workflow report of a protected order or workflow,
- open a change request that contains a protected object,
- see any transaction history data created by a protected order or workflow.

In case several confidential objects are linked with each other (e.g. a confidential building block is used in a recipe or a confidential workflow is assigned to an order), only the access privilege defined for the master object is required to access the linked objects with the master object as starting point. In order to link two confidential objects, they do not need to be protected by the same access privilege. The user, however, needs to have both access privileges.
Example scenario: A workflow and an order have different access privileges. A user with access to both can assign the workflow to the order. Afterwards another user who only has the access privilege of the order is able to print the batch report and can review exceptions of the workflow if the review is started with the order as starting point.

STRUCTURE OF THE DOCUMENT

This description is structured in the following sections:

Recipe Designer (SR3071.6+)

Access control (SR3071.6.2)

The access control of the Recipe Designer workbench enables the definition of user-specific access rights.
Upon user login, the Recipe Designer workbench checks if a user is authorized to edit recipes. If not, the system provides restricted access that does not allow to save manipulated data. Only status transitions of master recipes, custom building blocks, and change requests can be triggered by authorized users.

Workflow Designer (SR3071.7+)

Access control (SR3071.7.2)

The access control of the Workflow Designer workbench enables the definition of user-specific access rights.
Upon user login, the Workflow Designer workbench checks if a user is authorized to edit workflows. If not, the system provides restricted access that does not allow to save manipulated data. Only status transitions of master workflows, custom building blocks, and change requests can be triggered by authorized users.

Chapter 3

Management of Master Recipes and Master Workflows (SR3146.4+)

- As requirements have been restructured compared to the original DRS document, this section also contains requirements of other requirement groups than SR3146.4+.

This section provides the requirements relevant to the management of master recipes and master workflows:

A **master recipe** has relationships to the following objects:

- Objects to build a procedural structure - building blocks (page 14): Procedure, unit procedure, operation, phase
- Capabilities to control the behavior during execution
  Detachable - Operation (SR3146.9.5.9.1) capability (page 23)
  Event-triggered (SR3146.9.5.9.2) capability (page 24)
  Escalation-enabled (SR3146.9.5.9.3) capability (page 24)
  Trigger-enabled (SR3146.9.5.9.4) capability (page 25)
  Server-run (SR3146.9.5.9.5) capability (page 25)
  Auto-startable (SR3146.9.5.9.6) capability (page 25)
  Pause-enabled (SR3146.9.5.10.1) capability (page 26)
  Detachable - Unit procedure (SR3146.9.5.10.2) capability (page 26)
  Abort-and-reactivate-enabled (SR3146.9.5.10.3) capability (page 27)
- Resource master data
  Material (**process input (SR3146.9.5.1)** data (page 18), **process output (SR3146.9.6.1)** data (page 27))
  Equipment requirement (class) (SR3146.9.5.5) data (page 21)
  Work center (SR3146.9.5.4) data (page 21)
  Station (SR3146.9.5.8) data (page 21)
- **Material flow control (SR3146.9.8)** data (page 43)
- **Signature privilege (SR3146.9.5.3)** data (page 20)

The diagram illustrates the connections between the objects.

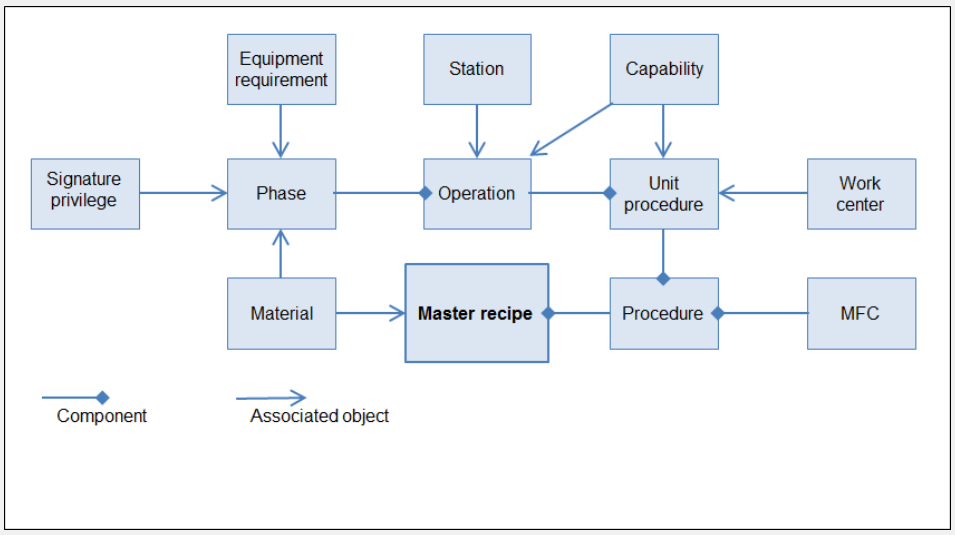


Figure 1: Objects relevant to recipe management

A **master workflow** has relationships to the following objects:

- Objects to build a procedural structure - building blocks (page 14):
  Procedure, unit procedure, operation, phase
- Resource master data
  Material (**process input (SR3146.9.5.1)** data (page 18), **process output (SR3146.9.6.1)** data (page 27))
  Equipment requirement (class) (SR3146.9.5.5) data (page 21)
  Work center (SR3146.9.5.4) data (page 21)
  Station (SR3146.9.5.8) data (page 21)
- Capabilities to control the behavior during execution
  Detachable - Operation (SR3146.9.5.9.1) capability (page 23)
  Event-triggered (SR3146.9.5.9.2) capability (page 24)
  Escalation-enabled (SR3146.9.5.9.3) capability (page 24)
  Trigger-enabled (SR3146.9.5.9.4) capability (page 25)
  Server-run (SR3146.9.5.9.5) capability (page 25)
  Auto-startable (SR3146.9.5.9.6) capability (page 25)
  Pause-enabled (SR3146.9.5.10.1) capability (page 26)
  Detachable - Unit procedure (SR3146.9.5.10.2) capability (page 26)
- **Material flow control (SR3146.9.8)** data (page 43)
- **Signature privilege (SR3146.9.5.3)** data (page 20)

The diagram illustrates the connections between the objects.

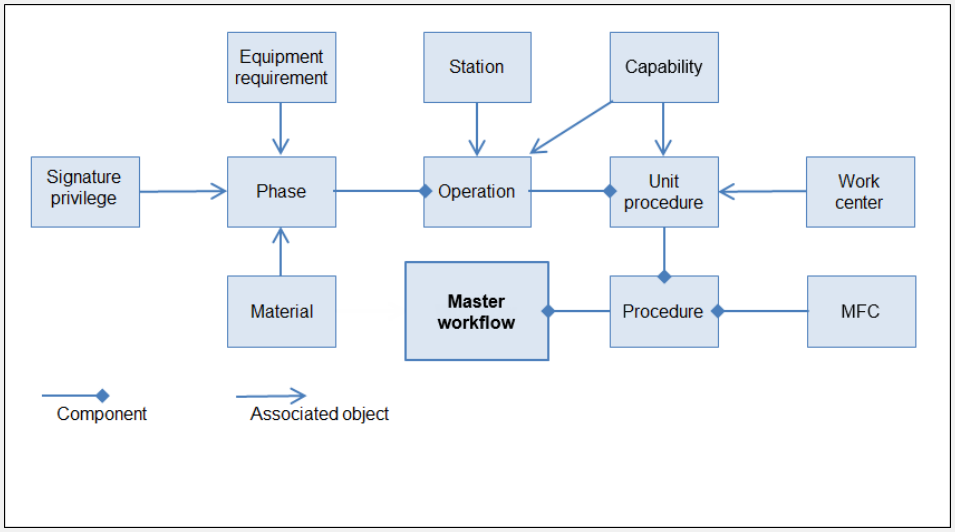


Figure 2: Objects relevant to workflow management

Master Recipe Model - S88 (SR3146.9.3)

The applied master recipe model is S88-compliant and includes

- **header attribute (SR3146.9.3.1)** data (page 10),
- a **procedural structure (SR3146.9.4)** hierarchy (page 13), and
- the following formula-related data:
  - **process input (SR3146.9.5)** data (page 18),
  - **process output (SR3146.9.6+)** data (page 27),
  - **process parameter (SR3146.9.7+)** data (page 29).

The model also applies to the respective top-level custom BB in support of **BB management (SR3146.10+)** capabilities (page 83).

Master Recipe - Header Attributes (SR3146.9.3.1)

The master recipe object supports the following header attributes:

- Basic data:
  - Identifier
  - Version
  - Description
  - Comparison baseline
    Version is displayed as <major revision>.x.
  - Method
  - Method description
  - Reason for creation
  - Usage type (Production, Cost center)
  - Access privilege
- Status data:
  - Status
  - Valid from/until

- Additional data:
  - Planned quantity
  - Minimum quantity
  - Maximum quantity
  - Registration number
  - Review mode (Automatic, Manual) (page 11)
  - Comment
- Material (product):
  - Material identifier
  - [Material] Short description
  - ERP BOM alternative
  - ERP BOM base quantity
  - ERP BOM released
- Packaging Levels (editable with the **Packaging Level Data (SR3146.9.2.17.3)** editor (page 159)):
  - L6 (Meaning, Contained number, Inventory level, Hide during execution)
  - ...
  - L9 (Meaning, Contained number, Inventory level, Hide during execution)

The attributes are maintained in the **header component-specific (SR3146.9.2.15)** property window (page 144).

Master Recipe - Review Mode (SR3146.9.3.2)

The system allows to define how an order that is based on a master recipe can be reviewed:

- Automatic
  The order will be reviewed automatically.

  The status of the order is automatically changed to **Reviewed**, once the order has been set to **Finished** and no exceptions with a risk other than **None** have been recorded for the order or any of its appended workflows.
  See also **Automatic Review of an Order (SR1084.4.1.1)** in "Functional Requirement Specification Runtime Data Management" [A3] (page 171).
- Manual
  The order should be reviewed manually.

Master Workflow Model - S88 (SR3146.9.12)

The applied master workflow model is S88-compliant and includes

- **header attribute (SR3146.9.12.1)** data (page 12),
- a **procedural structure (SR3146.9.4)** hierarchy (page 13), and
- the following formula-related data:
  - **process input (SR3146.9.5)** data (page 18),
  - **process output (SR3146.9.6+)** data (page 27),
  - **process parameter (SR3146.9.7+)** data (page 29).

The model also applies to the respective top-level custom BB in support of **BB management (SR3146.10+)** capabilities (page 83).

Master Workflow - Header Attributes (SR3146.9.12.1)

The master workflow object supports the following header attributes:

- Basic data:
  - Identifier
  - Version
  - Processing name
    Preset with the master workflow identifier. Once the Processing name differs from the workflow identifier, the system does not change it automatically when the master workflow is copied or renamed.
  - Short description
  - Description
  - Comparison baseline
    Version is displayed as <major revision>.x.
  - Reason for creation
  - Access privilege
- Status data:
  - Status
  - Valid from/until
- Additional data:
  - Review mode (Automatic, Manual) (page 13)
  - Type
  - One-click startable

- Execution prefix
- Production-relevant
- Appendable during processing
- Comment

The attributes are maintained in the **header component-specific (SR3146.9.2.15)** property window (page 144).

The **Type** attribute defines to which group the one-click startable workflow belongs in the Cockpit of the Production Execution Client. See also **Grouping (SR3071.4.1.1.4)** in "Functional Requirement Specification Execution Framework" [A2] (page 171).

The **Appendable during processing** attribute can only be set if the **Production-relevant** attribute is set.

Master Workflow - Review Mode (SR3146.9.12.2)

The system allows to define how a workflow can be reviewed:

- Automatic
  The workflow be reviewed automatically.

  The status of the workflow is automatically changed to **Reviewed**, once the workflow has been set to **Finished** and no exceptions with a risk level other than **None** have been recorded for the workflow.
  See also **Automatic Review of an Order (SR1084.4.1.1)** in "Functional Requirement Specification Runtime Data Management" [A3] (page 171).
- Manual
  The workflow should be reviewed manually.

Procedural Structure (SR3146.9.4)

The supported procedural structure includes

- Building blocks (page 14)
  (procedure, unit procedure, operation, phase)
- Links and branches (page 15)
  (connector, loop, selection branch, simultaneous branch)
- Transitions (page 15)

For master workflows, only one procedure and one unit procedure can be defined.

Procedural structure - Building blocks

The building blocks are:

- Procedure
  The procedure is the highest level in the (recipe) hierarchy and defines the strategy for carrying out a major processing action such as making a batch. It is defined in terms of an ordered set of unit procedures.
- Unit procedure
  According to S88, a unit procedure consists of an ordered set of operations that causes a contiguous production sequence to take place within a unit. Only one operation is presumed to be active in a unit at any time. An operation is carried to completion in a single unit. However, multiple unit procedures of one procedure may run concurrently, each in different units.
  However, PharmaSuite allows modeling and execution of parallel operations in compliance with general SFC rules.
  See also "Procedural Structure - Pre-defined Texts for Exceptions and Comments (SR3146.9.4.6)" (page 17).
- Operation
  An operation is an ordered set of phases that defines a major processing sequence that takes the material being processed from one state to another, usually involving a chemical or physical change. It is often desirable to locate operation boundaries at points in the procedure where normal processing can safely be suspended.
  See also "Procedural Structure - Operation Type (SR3146.9.4.4)" (page 16) and "Procedural Structure - Pre-defined Texts for Exceptions and Comments (SR3146.9.4.6)" (page 17).
- Phase
  The smallest element of procedural control that can accomplish a process-oriented task is a phase.
  See also "Procedural Structure - Pre-defined Texts for Exceptions and Comments (SR3146.9.4.6)" (page 17).

Procedural structure - Links and Branches

The Designer workbench uses various types of links to connect building blocks:

- Connector
  A link is a simple connector between two components.
- Loop
  A loop is a specific type of link that points back to a previous step in the graph. The loop endpoint must be on the same sequence as the loop start point, so you cannot draw a loop out of a branch.
- Selection branch
  A selection branch opens and closes two or more exclusive alternative sequences of steps in the graph.
  During execution, only one of the sequences must be executed before an operator can proceed with the step that follows after the sequences have been joined again. The result of the transition conditions that precede a branch determines the subsequent step.
- Simultaneous branch
  A simultaneous branch opens and closes two or more parallel sequences of steps in the graph. The leftmost sequence in the graph is the first to be displayed during execution according to the default execution order of SFCs (left to right).
  During execution, all parallel sequences must be executed before an operator can proceed with the step that follows after the sequences have been joined again.

Procedural structure - Transitions

A transition consists of a condition that defines which step is the next to be executed when there is more than one potential successor step. Thus, a transition predetermines which step to choose in a selection branch during execution or whether a loop needs to be executed. Transitions with conditions and or a description are equipped with transition identifiers. Any transition can be locked. For details about transition conditions, see "Procedural Structure - Conditions (SR3146.9.4.2)" (page 16), "Locking of Transitions (SR3146.9.14+)" (page 32), and "Expressions for Transitions (SR3146.9.9.4.2)" (page 150).

Procedural structure - Parallel execution (SR3146.9.4.1)

Building blocks can be executed in parallel.

Procedural structure - Conditions (SR3146.9.4.2)

Transitions can include conditions based on calculations.

- Transition conditions are based on the results of a previous step.
- Each transition is preset with a default condition, which defines that the step(s) preceding a transition must have been finished before the step(s) after the transition can be started.
  Transition conditions can be edited in an **Expression editor (SR3146.9.9.4)** UI element (page 148).
- Non-default conditions are marked by an indicator.

Procedural structure - Hidden phase (SR3146.9.4.5)

During recipe or workflow authoring, the system provides a Hidden Phase, which is a structural phase that is required to build adjacent branches or loops in valid SFC syntax. Thus, it is only visible in Recipe and Workflow Designer and does not appear as executable phase during order or workflow processing.
Two different Hidden Phases are available, one for the execution on a client and one for the execution on a server.

Procedural structure - Operation type (SR3146.9.4.4)

➢ Does not apply to Workflow Designer.

Operations are equipped with a **Dispense** attribute that marks an operation as being intended for dispensing if set. The recipe author has to define the operation type.

- The default setting is **non-Dispense**.
- The **Dispense** attribute is automatically populated to the superordinate unit procedure. For unit procedures, the **Dispense** attribute is read-only.

The following rules apply:

- A Dispense operation cannot be combined with other non-Dispense operations.
- A Dispense unit procedure must contain exactly one Dispense operation.

Procedural structure - Pre-defined texts for exceptions and comments (SR3146.9.4.6)

Phases, operations, and unit procedures are equipped with the Operator exception texts and Reviewer exception texts attributes. They allow to select two lists of pre-defined texts to be appended to the exception description and exception comment description during execution and review by exception. The recipe or workflow author must select the lists to be used.

By default, no lists are defined.

The attributes are maintained in the **element component-specific (SR3146.9.2.12)** property window (page 144).

Procedural structure - Hide in batch/workflow report (SR3146.9.4.7)

Phases can be flagged to not appear with their sub-reports in the batch or workflow report. This can shorten the size of reports.
In case the phase has an exception reference, the phase sub-report will be part of the report, regardless of the configuration.

By default, the flag is not set.

The attribute is maintained in the **element component-specific (SR3146.9.2.12)** property window (page 144).

Procedural structure - Group of Building Blocks (SR3146.9.4.8)

Unit procedures and operations can be foreseen to be used as groups of building blocks, a unit procedure as an operation group and an operation as a phase group. During custom building block authoring, the recipe author defines if a group usage is possible.
Allowing a custom building block to be used as a group increases the reusability for this building block. The building block can be used as single element on its own level or as group of building blocks on the next lower level. The concept allows to approve building blocks with a setup how it is used that applies to most cases. Once used, further elements can be added before, after, and parallel to the group without having to approve the group part again or to increase the amount of unit procedures or operations, respectively, which may require further configuration or instructions for the executing operator to handle them correctly, when running unit procedures at the same work center.
During processing on the shop floor, the group usage is not visible. An element of a group is executed in the same way as a single element.

Process Inputs (SR3146.9.5)

According to S88, process inputs are supported as part of the formula.

The attributes of process input parameters are maintained in the **Parameter Panel (SR3146.9.2.7)** UI element (page 142).

Process Inputs - Material (SR3146.9.5.1)

Materials are supported as process input parameters.

The minimum and maximum numbers of process inputs are taken over from the phase building block.

Process Inputs - Material Attributes (SR3146.9.5.1.1)

The material input parameters support the following attributes:

- Basic Data
  - Lock
  - Material identifier (read-only)
  - [Material] Short description (read-only)
  - Planned quantity mode (None, As defined, As produced)
  - Planned quantity (with UoM)
  - Usage type (e.g. Input, Output) (read-only)
  - Usage name
  - MFC-relevant (Yes/No), default = No
  - Position
  - Weighing sequence
  - Hierarchy level (read-only)
  - Path (read-only)
- Weighing and Production
  - Default weighing method
  - Allowed weighing methods
  - Tolerances (lower and upper, absolute and relative)
  - Target sublot status (---, Blocked, Quality Inspection, Unrestricted)
  - Weighing material type

- Planned potency
- Fixed quantity (Yes/No)
- Target weight
  (not relevant to material inputs and ignored during execution)
- Target weight tolerances (lower and upper, absolute and relative)
  (not relevant to material inputs and ignored during execution)

- Packaging Levels (editable with the **Packaging Level Data (SR3146.9.2.17.3)** editor (page 159)):
  (not relevant to material inputs and ignored during execution)
  - L0 (Meaning, Contained number, Inventory level, Hide during execution)
    (the Contained number attribute is not relevant to the L0 level)
  - ...
  - L5 (Meaning, Contained number, Inventory level, Hide during execution)

- The **Basic** attributes are displayed in both the horizontal and vertical panes of the **Parameter Panel (SR3146.9.2.7)** UI element (page 142).
- Materials can be flagged as MFC-relevant and thus be available for MFC management (page 43).
- Weighing-specific attributes can be defined.
  Weighing methods and tolerances are populated automatically from the material. Weighing material type and planned potency are preset with default values and will be populated into the respective list of materials item.
- Target sublot status is populated automatically from the material. As process input it defines the status of the temporary sublot created at batch identification.
- Positions of MFC-relevant materials (process input) can be defined.
- For master recipes based on an ERP BOM, the ERP BOM items can be assigned as material parameters via **Setlist (SR3146.9.2.9)** UI element (page 140).

Process Inputs - Signature Privilege (SR3146.9.5.3)

Signature privileges are supported as process input parameters.

- They apply to phase-specific exceptions, explicit capability-related exceptions, phase actions, and the phase completion.
- The correlation between signature privilege and exception-specific risk level must be defined manually.
- Per PharmaSuite installation, a default signature privilege can be configured which will be used if no signature privilege has been assigned to a phase for a specific risk level.

Process Inputs - Signature Privilege Attributes (SR3146.9.5.3.1)

The signature privilege input parameters support the following attributes:

- Read-only:
  - Identifier
  - Description (only in the **Universe (SR3146.9.2.4+)** UI element (page 137))
  - Signature type (Single/Double, only in the **Universe (SR3146.9.2.4+)** UI element (page 137))
  - Default reason (1st)
  - Default reason (2nd)
  - Category
- Editable:
  - Usage (High risk exception, High risk exception (comment mandatory), Medium risk exception, Medium risk exception (comment mandatory), Low risk exception, Low risk exception (comment mandatory), No risk exception, Phase completion, Phase completion (sequential), Phase action)
  - Usage name
  - Reason (1st)
    - Maximum length is 80 characters.
  - Reason (2nd)
    - Maximum length is 80 characters.

PROCESS INPUTS - WORK CENTER (SR3146.9.5.4)

Work centers are supported as process input parameters on unit procedure level.

The operations and phases of the unit procedure have to be processed at one of the defined work centers.

PROCESS INPUTS - WORK CENTER ATTRIBUTES (SR3146.9.5.4.1)

The work center input parameters support the following attributes:

- Identifier
- Description
- Storage area

PROCESS INPUTS - STATION (SR3146.9.5.8)

Stations are supported as process input parameters on operation level.

The phases of the operation have to be processed at one of the defined stations.

PROCESS INPUTS - STATION ATTRIBUTES (SR3146.9.5.8.1)

The station input parameters support the following attributes:

- Identifier
- Description
- Work center

PROCESS INPUTS - EQUIPMENT REQUIREMENT (CLASS) (SR3146.9.5.5)

Equipment requirements of the **Class** type are supported as process input parameters.

PROCESS INPUTS - REQUIREMENT ATTRIBUTES (CLASS) (SR3146.9.5.5.1)

The equipment class input parameters support the following attributes:

- Read-only:
  - Identifier
  - Short description
  - Description
  - Level
  - Type (e.g. Class, Item)

- Editable:
  - BOE position
  - Usage name

PROCESS INPUTS - EQUIPMENT REQUIREMENT (PROPERTY TYPE) (SR3146.9.5.6)

Property types are supported as part of an already existing equipment requirement (e.g. equipment class).

PROCESS INPUTS - REQUIREMENT ATTRIBUTES (PROPERTY TYPE) (SR3146.9.5.6.1)

The (equipment) property type input parameters support the following attributes:

- Editable:
  - Identifier (unique)
  - Description
  - Rule (only editable for property types of the **Specification** usage type and of the **Runtime** usage type, not editable for property types of the **Automation** or **Historian** usage type, representation always shows the related property type identifier)
- Read-only:
  - Type (Dependent, Independent)

PROCESS INPUTS - EQUIPMENT REQUIREMENT (RULE) (SR3146.9.5.7)

Rules are supported as part of an already existing equipment requirement.
Flexible rules apply to a single equipment entity and the parent entity of an equipment entity group whereas conditional rules apply to a single equipment entity and to all entities of an equipment entity group.

PROCESS INPUTS - REQUIREMENT ATTRIBUTES (RULE) (SR3146.9.5.7.1)

The rules of an equipment class input parameter supports the following attributes:

- Editable (with the **Expression editor (SR3146.9.9.4)** UI element (page 148)):
  - Identifier (unique)
  - Description
  - Rule
- Read-only:
  - Type (Flexible, Conditional rule (group-enabled))

PROCESS INPUTS - TRANSITION (SR3146.9.5.11)

Transitions are supported as process input parameters.

PROCESS INPUTS - TRANSITION ATTRIBUTES (SR3146.9.5.11.1)

The transition input parameters support the following attributes:

- Identifier
- Description
- Condition

Transition conditions can be edited in an **Expression editor (SR3146.9.9.4)** UI element (page 148).

PROCESS INPUTS - CAPABILITIES ON OPERATION LEVEL (SR3146.9.5.9)

Capabilities are supported as process input parameters for operations.

PROCESS INPUTS - DETACHABLE CAPABILITY - OPERATION (SR3146.9.5.9.1)

The system provides the Detachable capability, which supports the following attributes:

- Read-only:
  - Identifier
  - Action
- Editable:
  - Exception enabled
  - Risk assessment
  - Exception text

This capability allows an operator to detach a running operations from its work center and station (see **Detaching an Operation (SR1089.6.1+)** in "Functional Requirement Specification Execution Framework" [A2] (page 171)).

This capability cannot be used in Dispense operations.

PROCESS INPUTS - EVENT-TRIGGERED CAPABILITY (SR3146.9.5.9.2)

The system provides the Event-triggered capability, which supports the following attributes:

- Read-only:
  - Identifier
  - Action (New run, Cancel run, Cancel all runs, Remove template)
- Editable (for each action):
  - Exception enabled
  - Risk assessment
  - Exception text

This capability cannot be used in Dispense operations.

PROCESS INPUTS - ESCALATION-ENABLED CAPABILITY (SR3146.9.5.9.3)

The system provides the Escalation-enabled capability, which supports the following attributes:

- Read-only:
  - Identifier
  - System action (Run created, Run due reminder, Run overdue, Run expired, Finish due reminder, Finish overdue)
- Editable:
  - Duration before action (since trigger) (not editable for the **Run created** system action)
  - Automatic exception enabled (not editable for the **Run created** system action)
  - Risk assessment (not editable for the **Run created** system action)
  - Exception text (not editable for the **Run created** system action)
  - Alarm enabled
  - Notification text

This capability is only supported for event-triggered operations.
This capability cannot be used in Dispense operations.

It is not expected that HTML tags are used as part of the notification or exception text. This means that the notification text will be shown as plain text and HTML formatting will be not interpreted by the notification panel.

Process Inputs - Trigger-enabled Capability (SR3146.9.5.9.4)

The system provides the Trigger-enabled capability, which supports the following attributes:

- Read-only:
  - Identifier
- Editable (for each new line item):
  - Trigger (unique)
  - Description
  - Trigger phase

The system allows to add multiple line items for multiple trigger references.

This capability is only supported for event-triggered operations.
This capability cannot be used in Dispense operations.

Process Inputs - Server-run Capability (SR3146.9.5.9.5)

The system provides the Server-run capability, which supports the following attributes:

- Read-only:
  - Identifier

This capability cannot be combined with other capabilities.

This capability cannot be used in Dispense operations.

Process Inputs - Auto-startable Capability (SR3146.9.5.9.6)

The system provides the Auto-startable capability, which supports the following attributes:

- Read-only:
  - Identifier

This capability cannot be used in Dispense operations.

Process Inputs - Capabilities on Unit Procedure Level (SR3146.9.5.10)

Capabilities are supported as process input parameters for unit procedures.

PROCESS INPUTS - PAUSE-ENABLED CAPABILITY (SR3146.9.5.10.1)

The system provides the Pause-enabled capability, which supports the following attributes:

- Read-only:
  - Identifier
  - Action
- Editable:
  - Exception enabled
  - Risk assessment
  - Exception text

PROCESS INPUTS - DETACHABLE CAPABILITY - UNIT PROCEDURE (SR3146.9.5.10.2)

The system provides the Detachable capability, which supports the following attributes:

- Read-only:
  - Identifier
  - Action
- Editable:
  - Exception enabled
  - Risk assessment
  - Exception text

This capability allows an operator to detach a running unit procedure from its work center and station (see **Detaching a Unit Procedure (SR1089.8.4+)** in "Functional Requirement Specification Execution Framework" [A2] (page 171)).

This capability cannot be used in Dispense unit procedures.

PROCESS INPUTS - ABORT-AND-REACTIVATE-ENABLED CAPABILITY (SR3146.9.5.10.3)

The system provides the Abort-and reactivate-enabled capability, which supports the following attributes:

- Read-only:
  - Identifier
  - Action
- Editable:
  - Exception enabled
  - Risk assessment
  - Exception text

This capability allows an operator to abort and reactivate a running unit procedure (see **Reactivating a Unit Procedure (SR1089.8.5)** in "Functional Requirement Specification Execution Framework" [A2] (page 171)).

This capability cannot be used in Dispense unit procedures and master workflows.

Process Outputs (SR3146.9.6+)

According to S88, process outputs are supported as part of the formula.

The attributes of process output parameters are maintained in the **Parameter Panel (SR3146.9.2.7)** UI element (page 142).

PROCESS OUTPUTS - MATERIAL (SR3146.9.6.1)

Materials in the sense of product are supported as process output parameters.

The minimum and maximum numbers of process outputs are taken over from the phase building block.

PROCESS OUTPUTS - MATERIAL ATTRIBUTES (SR3146.9.6.1.1)

The material output parameters support the following attributes:

- Basic Data
  - Lock
  - Material identifier (read-only)
  - [Material] Short description (read-only)

- Planned quantity mode (None, As defined)
- Planned quantity (with UoM)
- Usage type (e.g. Input, Output) (read-only)
- Usage name
- MFC-relevant (Yes/No), default = No
- Position
- Hierarchy level (read-only)
- Path (read-only)

- Weighing and Production
  - Default weighing method
  - Allowed weighing methods
  - Tolerances (lower and upper, absolute and relative)
  - Target sublot status (---, Blocked, Quality Inspection, Unrestricted)
  - Weighing material type
    (not relevant to material outputs and ignored during execution)
  - Planned potency
    (not relevant to material outputs and ignored during execution)
  - Fixed quantity (Yes/No)
    (not relevant to material outputs and ignored during execution)
  - Target weight
  - Target weight tolerances (lower and upper, absolute and relative)
- Packaging Levels (editable with the **Packaging Level Data (SR3146.9.2.17.3)** editor (page 159)):
  - L0 (Meaning, Contained number, Inventory level, Hide during execution)
    (the Contained number attribute is not relevant to the L0 level)
  - ...
  - L5 (Meaning, Contained number, Inventory level, Hide during execution)

- Materials can be flagged as MFC-relevant and thus be available for MFC management (page 43).
- Target sublot status is populated automatically from the material.

Process Parameters (SR3146.9.7+)

According to S88, process parameters are supported as part of the formula.

- Process parameters and their attributes are taken over from the phase building block and parameter class, respectively. Except for the **Instruction table (SR3146.9.7.5)** process parameters (page 30) and the **Instruction link (SR3146.9.7.6)** process parameters (page 31), which are provided by the framework and are available for each phase building block.
- For phases located within the same operation and across operations, the output of a phase can be used as input of a parameter attribute of another phase.
- Inputs can be edited by an **Expression editor (SR3146.9.9.4)** UI element (page 148).
- The phase that provides the output must have been processed before the process step that uses the output as input.
  Phases and operations may or may not be passed (e.g. in simultaneous branches). If they are not passed, the output is Null. This needs to be reflected in the referencing expression.

Process Parameters - Classification (SR3146.9.7.1)

For process parameters, the following classifications are available:

- Critical quality attribute
- Critical process parameter
- Key quality attribute
- Key process parameter
- General quality attribute
- General process parameter

Dynamic Creation of Process Parameters (SR3146.9.7.4)

The **Parameter Panel (SR3146.9.2.7)** UI element (page 142) for process parameters supports the dynamic creation of process parameters for specific phases during recipe and workflow authoring. The availability of a parameter is controlled by the current phase.

When a new parameter bundle is created, the recipe or workflow author defines the bundle identifier.

Process parameters that have been created in this way can also be deleted within the Parameter Panel.

Bundle Process Parameters (SR3146.9.7.4.1)

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

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

Bundle Output Variables (SR3146.9.7.4.2)

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

The naming convention applies to PharmaSuite product phases.

INSTRUCTION TABLE PROCESS PARAMETERS (SR3146.9.7.5)

The **Parameter Panel (SR3146.9.2.7)** UI element (page 142) for process parameters provides instruction table process parameters for each phase during recipe and workflow authoring as framework parameter bundle.

When a new instruction table is created, the recipe or workflow author defines the identifier of the process parameter. A maximum of 10 instruction tables with 50 rows per table is supported. The instruction table is displayed in the **Phase Preview (SR3146.9.2.11)** UI element (page 148) in Recipe and Workflow Designer. It is not displayed in the **Preview mode** of a phase building block.
For the definition-specific process parameter of an instruction table, its internal identifier is populated from the initial table identifier.
For all text-specific process parameters of an instruction table, their internal identifiers are a concatenation of the internal table identifier and text identifier.

The sequence of text-specific process parameters of an instruction table can be changed.

Instruction table process parameters that have been created in this way can also be deleted within the Parameter Panel.

For server-run phases, instruction table process parameters cannot be created since the phases do not have a graphical representation (UI).

Instruction Link Process Parameters (SR3146.9.7.6)

The **Parameter Panel (SR3146.9.2.7)** UI element (page 142) for process parameters provides instruction link process parameters for each phase during recipe and workflow authoring as framework parameter bundle.

When a new instruction link is created, the recipe or workflow author defines the identifier of the process parameter. A maximum of 10 instruction link parameters per phase is supported. Each instruction link parameter supports a maximum of 10 instruction link URLs. The instruction links are displayed in the **Phase Preview (SR3146.9.2.11)** UI element (page 148) in Recipe and Workflow Designer. They are not displayed in the **Preview mode** of a phase building block.

Instruction link process parameters that have been created in this way can also be deleted within the Parameter Panel.

For server-run phases, instruction link process parameters cannot be created since the phases do not have a graphical representation (UI).

Locking of Parameters (SR3146.9.13+)

Locking and unlocking of parameters is supported.

Locked Parameters (SR3146.9.13.1)

Parameters can be locked and unlocked as long as the related master recipe or master workflow is not in the **Verification** or **Valid (SR3146.6.4)** statuses (page 49) or the custom BB is not in the **Verification** or **Approved (SR3146.10.1)** statuses (page 87). The attributes of a locked parameter are read-only for this specific occurrence. They can again be updated when the parameter is unlocked.

Locking of Bundle Parameters (SR3146.9.13.3)

As soon as at least one process parameter of a **bundle process parameter (SR3146.9.7.4.1)** parameter (page 30) is locked, the process parameter can no longer be removed.

Thus, unlocked bundle parameters still can be removed from and new bundle process parameters still can be added to a custom building block in the **Verification** or **Approved (SR3146.10.1)** statuses (page 87).

FROZEN PARAMETERS (SR3146.9.13.2)

A frozen parameter of a master recipe, master workflow, or custom BB is permanently locked if the related source custom BB was in the **Verification** or **Approved (SR3146.10.1)** statuses (page 87) at the time when it was copied for usage. Thus, the related attributes cannot be updated anymore.

Locking of Transitions (SR3146.9.14+)

Locking and unlocking of transitions is supported.

LOCKED TRANSITIONS (SR3146.9.14.1)

Transitions can be locked and unlocked as long as the related master recipe or master workflow is not in the **Verification** or **Valid (SR3146.6.4)** statuses (page 49) or the custom BB is not in the **Verification** or **Approved (SR3146.10.1)** statuses (page 87). The attributes of a locked transition are read-only for this specific occurrence. They can again be updated when the transition is unlocked. The identifier of a transition is always read-only regardless of whether the transition is locked or not, if the transition is part of a used custom building block in a read-only status.

FROZEN TRANSITIONS (SR3146.9.14.2)

A frozen transition of a master recipe, master workflow, or custom BB is permanently locked if the related source custom BB was in the **Verification** or **Approved (SR3146.10.1)** statuses (page 87) at the time when it was copied for usage. Thus, the related attributes cannot be updated anymore.

Workbench Object Operations (SR3146.9.10+)

The Designer workbench supports specific object operations for:

- master recipe (SR3146.9.10.1) objects (page 33),
- master workflows (SR3146.9.10.9) objects (page 34),
- element (SR3146.9.10.6) objects (page 36),
- custom building block (SR3146.9.10.7) objects (page 84), and
- change request (SR3146.12+) objects (page 97).

Operations for links and transitions (page 37) are also supported.

Workbench Object Operations - Master Recipe (SR3146.9.10.1)

The following operations are supported for master recipes:

- Create new (product material-related) master recipe (page 34)
- Open existing (see **Open (SR3146.9.2.20+)** dialog (page 136))
- View locked master recipe (page 34)
- Save (current or all) master recipe
- Copy (Save as...)
  For master recipes that are protected by an access privilege: The access privilege of the original master recipe is passed on to the new master recipe. In the default configuration, the access privilege cannot be changed.
- Rename
- Delete
- Create building block from selected object
  For master recipes that are protected by an access privilege: The access privilege of the original master recipe is passed on to the new building block. In the default configuration, the access privilege cannot be changed.
- Insert existing procedure BB
- Replace Material (Product) (SR3146.9.10.11) (page 42)
- Status-related operations (see Version Control (SR3146.4.5) (page 161))
- Report (see Master Recipe Report - Batch (SR3146.11+) (page 115))
- Statistics (SR3146.9.2.18) (page 161)
- Close (current or all)

Master recipes can either be saved under a new identifier or under the same identifier and a new version number.

WORKBENCH OBJECT OPERATIONS - NEW MASTER RECIPE DIALOG (SR3146.9.10.1.2)

When a new master recipe is created, first, the recipe author defines the product material of the master recipe. The list of available materials shall be restricted by the access privileges of the logged-in user. The material identifier is populated as master recipe identifier. The master recipe identifier is editable.
The system shall allow to select an access privilege for the protection of the master recipe from unauthorized access for users with access privileges that allow to maintain protected recipes. The list of available access privileges shall be restricted to the access privileges of the logged-in user.
If an access privilege is defined for the material, this access privilege is passed on to the new master recipe. In the default configuration, the access privilege cannot be changed.

For master recipes based on an ERP BOM, the ERP BOM is selected from the **library for materials (ERP BOM) (SR3146.9.2.4.2)** UI element (page 139).
The **material process input (SR3146.9.5.1)** data (page 18) defined within Recipe Designer must match the materials, positions, and quantities set down in the ERP BOM. The ERP BOM items are available via **Setlist (SR3146.9.2.9.1)** UI element (page 141).

WORKBENCH OBJECT OPERATIONS - VIEW LOCKED MASTER RECIPE (SR3146.9.10.1.4)

If a master recipe is locked by another user, the system allows to view the master recipe without the possibility to change it.

The master recipe can be copied (save as). Operations that do not change a master recipe are also available (e.g. status history, print report, graph pagination, statistics).

Workbench Object Operations - Master Workflow (SR3146.9.10.9)

The following operations are supported for master workflows:

- Create new master workflow (page 35)
- Open existing (see **Open (SR3146.9.2.20+)** dialog (page 136))
- View locked master workflow (page 35)
- Save (current or all) master workflow
- Copy (Save as...)
  For master workflows that are protected by an access privilege: The access privilege of the original master workflow is passed on the new master workflow. In the default configuration, the access privilege cannot be changed.
- Rename
- Delete

- Create building block from selected object
  For master workflows that are protected by an access privilege: The access privilege of the original master workflow is passed on to the new building block. In the default configuration, the access privilege cannot be changed.
- Insert existing procedure BB
- Status-related operations (see Version Control (SR3146.4.5) (page 161))
- Master Workflow Report (SR3146.13+) (page 123)
- Statistics (SR3146.9.2.18) (page 161)
- Close (current or all)

Master workflows can either be saved under a new identifier or under the same identifier and a new version number.

Workbench Object Operations - New Master Workflow Dialog (SR3146.9.10.9.1)

When a new master workflow is created, first, the workflow author defines the master workflow identifier.
The system shall allow to select an access privilege for the protection of the master workflow from unauthorized access for users with access privileges that allow to maintain protected workflows. The list of available access privileges shall be restricted to the access privileges of the logged-in user.

Workbench Object Operations - View Locked Master Workflow (SR3146.9.10.9.3)

If a master workflow is locked by another user, the system allows to view the master workflow without the possibility to change it.

The master workflow can be copied (save as). Operations that do not change a master workflow are also available (e.g. status history, print report, graph pagination, statistics).

Workbench Object Operations - Element (SR3146.9.10.6)

The following operations are supported for elements that are derived from building blocks (e.g. recipe element within a recipe, workflow element within a workflow, BB element within a custom BB). They apply to elements on all levels (procedure, unit procedure, operation, phase).

- Create new (e.g. dummy phase)
- Insert element from an existing BB or from the Graph Clipboard via the **Setlist (SR3146.9.2.9)** UI element (page 140)
  It is not possible to insert a group of building blocks into its own graph.
- Copy selected element(s), see **Copy & Paste Function (SR3146.9.1.6+)** feature (page 133)
- Paste previously copied element(s), see **Copy & Paste Function (SR3146.9.1.6+)** feature (page 133)
- Create <same BB level> from selected element
  For elements that are part of a master recipe, master workflow, or custom building block that is protected by an access privilege: The access privilege of the header object is passed on to the new building block. In the default configuration, the access privilege cannot be changed.
- Create <next higher BB level> from selected element(s)
  For elements that are part of a master recipe, master workflow, or custom building block that is protected by an access privilege: The access privilege of the header object is passed on to the new building block. In the default configuration, the access privilege cannot be changed.
- Replace (overrides parameters) via the **Setlist (SR3146.9.2.9)** UI element (page 140)
- Smart replace via the **Setlist (SR3146.9.2.9)** UI element (page 140)
  (see **Workbench Object Operations - Smart Replace (SR3146.9.10.10)** operation (page 38))
- Rename
- Delete selected elements

- Connect elements manually. However, Recipe and Workflow Designer inserts possible or required transitions (page 15) and required selection/simultaneous branches (page 15) automatically.
- Insert branch with exclusive alternative steps (XOR). A selection branch (page 15) and required transitions (page 15) connect the elements.
- Insert loop. A loop, required selection branches, and a transition connect the elements.
- Insert branch with parallel steps (AND). A simultaneous branch (page 15) connects the elements.
- If a BB is created from a selected element, the new building block is available for selection from the **Universe (SR3146.9.2.4+)** UI element (page 137). It can be made available for the **Setlist (SR3146.9.2.9)** UI element (page 140) and can also be opened automatically in a new tab. Available revisions of custom building blocks starting with the same identifier are listed for information.
- Insert sequence. A link (page 15) and a transition (page 15) connect the elements.
- Select all elements and transitions along with or without links.
- Find elements by searching for a specific string.

Workbench Object Operations - Link and Transition

The following operations are supported for links (page 15) and transitions (page 15):

- Delete links and transitions selected in a graph.
- Select all links and transitions along with all elements in the current Graph Window.
- Select all transitions along with all elements in the current Graph Window.
- Select all links in the current Graph Window.

Workbench Object Operations - Smart Replace (SR3146.9.10.10)

With the **Smart replace** operation, building blocks (BBs) and elements of recipes, workflows, and BBs are replaced by other BBs from the Setlist. The operation differs from the **Replace** operation in its parameter and transition handling. It merges the parameter data and transitions of the replacing (new) building block or system phase into the parameter data and transitions of the (old) building block to be replaced. For matching, the system uses the identifiers that the elements have when the element (group) is moved to a read-only status. The same matching behavior applies for transitions.

A single building block can only be replaced by a group of building blocks and vice versa using the **Replace** operation, not the **Smart replace** operation.

If the new building block is a custom building block (cBB) and in the **Draft** or **Approved** status (page 87), the parameters are merged depending on their type, **locking state** (SR3146.9.13.1) (page 31), and existence; transitions are merged depending on their **locking state** (SR3146.9.14.1) (page 32) and existence.

- Merge strategy for **process parameter (SR3146.9.7+)** data (page 29) and their attributes

New cBB	Old cBB	Resulting building block after smart replace
Parameter is permanently locked (**Frozen**).	Parameter is permanently locked (**Frozen**).	The new parameter is **taken over**, it is still permanently locked (**Frozen**). The parameter of the old cBB is ignored.
	Parameter is **locked** for this specific occurrence on the BB, but could be unlocked.	
	Parameter is **not locked**.	
Parameter is **locked** for this specific occurrence on the BB, but could be unlocked.	Parameter is permanently locked (**Frozen**).	The new parameter is **taken over**, it is **locked** for this specific occurrence on the BB, but could be unlocked. The parameter of the old cBB is ignored. This combination is not relevant to **change request (SR3146.12+)** objects (page 97).
	Parameter is **locked** for this specific occurrence on the BB, but could be unlocked.	
	Parameter is **not locked**.	
Parameter is **not locked**.	Parameter is permanently locked (**Frozen**).	The previous parameter is **taken over**, but is **not locked**. The parameter of the new cBB is ignored.
	Parameter is **locked** for this specific occurrence on the BB, but could be unlocked.	The previous parameter is **taken over**, it is **still locked** for this specific occurrence on the BB, but could be unlocked. The parameter of the new cBB is ignored.

New cBB	Old cBB	Resulting building block after smart replace
	Parameter is **not locked**.	The previous parameter is **taken over**, it is still **not locked**. The parameter of the new cBB is ignored.
Parameter in any locking state.	No parameter present.	The new parameter is **taken over** in its locking state.
No parameter present.	Parameter in any locking state.	The previous parameter does no longer exist.

- Merge strategy for the following parameter-specific attributes:
  - The attributes are unique keys.
  - **Usage** attribute of **Signature privilege (SR3146.9.5.3)** parameters (page 20)
  - **BOE position** attribute of **Equipment requirement (class) (SR3146.9.5.5)** parameters (page 21)
  - **Work center identifier** attribute of **Work center (SR3146.9.5.4)** parameters (page 21)
  - **Station identifier** attribute of **Station (SR3146.9.5.8)** parameters (page 21)
  - **Capability identifier** attribute of **Capability (SR3146.9.5.9)** parameters (page 23)

New cBB	Old cBB	Resulting building block after smart replace
Parameter is permanently locked (**Frozen**).	Parameter is permanently locked (**Frozen**).	The new parameter is **taken over**, it is still permanently locked (**Frozen**). The parameter of the old cBB is ignored.
	Parameter is **locked** for this specific occurrence on the BB, but could be unlocked.	
	Parameter is **not locked**.	
	No parameter present.	The new parameter is **taken over**, it is still permanently locked (**Frozen**).
Parameter is **locked** for this specific occurrence on the BB, but could be unlocked.	Parameter is permanently locked (**Frozen**).	The new parameter is **taken over**, it is **still locked** for this specific occurrence on the BB, but could be unlocked. The parameter of the old cBB is ignored. This combination is not relevant to **change request (SR3146.12+)** objects (page 97).
	Parameter is **locked** for this specific occurrence on the BB, but could be unlocked.	
	Parameter is **not locked**.	

New cBB	Old cBB	Resulting building block after smart replace
	No parameter present.	The new parameter is **taken over**, it is **still locked** for this specific occurrence on the BB, but could be unlocked. This combination is not relevant to **change request (SR3146.12+)** objects (page 97).
Parameter is **not locked**.	Parameter is permanently locked (**Frozen**).	The previous parameter is **taken over**, but is **not locked**. The parameter of the new cBB is ignored.
	Parameter is **locked** for this specific occurrence on the BB, but could be unlocked.	The previous parameter is **taken over**, it is **locked** for this specific occurrence on the BB, but could be unlocked. The parameter of the new cBB is ignored.
	Parameter is **not locked**.	The previous parameter is **taken over**, it is still **not locked**. The parameter of the new cBB is ignored.
	No parameter present.	There is still no parameter present. The parameter of the new cBB is ignored.
No parameter present.	Parameter in any locking state.	The previous parameter is **still locked** (if locked before) and **not locked** in all other cases.

- Material parameters (**process input (SR3146.9.5.1)** data (page 18), **process output (SR3146.9.6.1)** data (page 27))
  They are completely ignored during the **Smart replace** operation.
- Merge strategy for transitions (page 23) and their attributes
  - The identifier is the unique key within a procedural element.

New cBB	Old cBB	Resulting building block after smart replace
Transition is permanently locked (**Frozen**).	Transition is permanently locked (**Frozen**).	The new transition is **taken over**, it is still permanently locked **(Frozen)**. The transition of the old cBB is ignored.
	Transition is **locked** for this specific occurrence on the BB, but could be unlocked.	
	Transition is **not locked**.	
Transition is **locked** for this specific occurrence on the BB, but could be unlocked.	Transition is permanently locked (**Frozen**).	The new transition is **taken over**, it is **locked** for this specific occurrence on the BB, but could be unlocked. The transition of the old cBB is ignored. This combination is not relevant to **change request (SR3146.12+)** objects (page 97).
	Transition is **locked** for this specific occurrence on the BB, but could be unlocked.	
	Transition is **not locked**.	

New cBB	Old cBB	Resulting building block after smart replace
Transition is **not locked**.	Transition is permanently locked (**Frozen**).	The previous transition is **taken over**, but is **not locked**. The transition of the new cBB is ignored.
	Transition is **locked** for this specific occurrence on the BB, but could be unlocked.	The previous transition is **taken over**, it is **still locked** for this specific occurrence on the BB, but could be unlocked. The transition of the new cBB is ignored.
	Transition is **not locked**.	The previous transition is **taken over**, it is still **not locked**. The transition of the new cBB is ignored.
Transition in any locking state.	No transition present.	The new transition is **taken over** in its locking state.
No transition present.	Transition in any locking state.	The previous transition does no longer exist.

Applies to all parameters and transitions:

- If a building block on a higher level (e.g. operation level) is replaced using the **Smart Replace** operation and the new cBB contains a new recipe element, workflow element, BB element, or transition that did not exist in the previous version, this new element or transition is taken over including all of its parameter and transition configurations.

Applies to material parameters:

- If a building block is replaced using the regular **Replace** operation or is added to a **procedural structure (SR3146.9.4)** hierarchy (page 13), all of its already assigned material parameters are still taken over. They are only ignored in case the **Smart replace** operation is used.

Applies to parameters if a system building block (phase) is replaced with the **Smart replace** operation:

- The new system phase cannot have locked parameters or bundle parameters. The system always retains the maintained parameters of the replaced element with their values. Only those process parameters (except bundle parameters) of the old system phase that are not present in the new system phase will not be part of the resulting system phase.

Applies to all bundle parameters of the old building block:

- The system does not differentiate between bundle parameters added manually to a phase after adding the building block containing this phase and bundle parameters that were already part of the building block before it was added. Bundle parameters of the old building block that are not present in the new building block will still be part of the resulting building block.

Workbench Object Operations - Replace Material (Product) (SR3146.9.10.11)

➢ Does not apply to Workflow Designer.

With the **Replace material (product)** operation, the material (product) of an existing master recipe, including its related ERP BOM (if applicable), can be replaced.

The system shall only allow to select ERP BOMs and materials whose access privilege matches the access privilege of the logged-in user.
If an access privilege is defined for the new ERP BOM or material that differs from the access privilege of the master recipe, the new access privilege is passed on to the master recipe.

Before the new material can be selected, the system requests a confirmation by the recipe author since the **Replace material (product)** operation can lead to a significant number of changes in the master recipe data.

Along with replacing the material (product), the system automatically updates the following data:

- Material (product) of the master recipe's header attributes
- Material of the final MFC output
- Material of the MFC-relevant material input and output parameters (that used the previous material)
- Packaging levels (Meaning, Contained number) of the material parameters (L0, …, L5) (that used the previous material)
- Only if the new material (product) is related to an ERP BOM:
  - ERP BOM items in the Setlist
  - Planned quantity of the master recipe
  - Packaging levels of the master recipe (L6, …, L9)
  - Material inputs by BOM position (material identifier, material short description, planned quantity, and fixed quantity)
  - In case linked MFC inputs/outputs, according to the MFC definition, refer to the same material:
    The system also updates the material identifier and planned quantity of the related material outputs (same unit procedure, e.g. Dispense, planned quantity only for the first linked output) and the material identifier of the subsequent material inputs (subsequent unit procedure, e.g. Mixing).

In case there are remaining material-related inconsistencies that cannot be resolved automatically, the system displays the corresponding messages in the **Messages (SR3146.9.2.16+)** window (page 144). For details, see **Validator Consistency Checks (SR3146.9.4.3+)**, Checks that Apply to Master Recipes Based on an ERP BOM (page 63) and Checks that Apply to Material Parameters of a Phase (page 65).

Details of the performed changes, are visible in the **Comparison (SR3146.9.2.19+)** window (page 146), based on a defined comparison baseline.

Material Flow Control (MFC) (SR3146.9.8)

A graphical tool allows to model the material flow of a master recipe or master workflow through a procedure within the Recipe and Workflow Designer workbench. The MFC-relevant material parameters (SR3146.9.5.1 (page 18), SR3146.9.6.1 (page 27)) provide the basis for the material flow.

Graphical View (SR3146.9.8.1)

The graphical representation displays the following components:

- Unit procedures (of the procedure)
- MFC-relevant material inputs
- MFC-relevant material outputs

- Unit procedures hold all MFC-relevant material nodes defined for their respective phases.
  Before a merge, there are no connections available between unit procedures.
- Direct MFC inputs are derived from the material input parameters of the procedure's phases.
  A direct MFC input is displayed as triangle with an outgoing connector.
  The number it shows corresponds to the numeric suffix of the MFC item identifier visible in the table view of the MFC data.
- Does not apply to Workflow Designer.
  Direct MFC outputs are derived from the material output parameters of the procedure's phases.
  A direct MFC output is displayed as triangle with an incoming connector.
  The number it shows corresponds to the numeric suffix of the MFC item identifier visible in the table view of the MFC data.
- Does not apply to Workflow Designer.
  Switch nodes are created when direct MFC inputs and outputs are merged. A direct MFC input is replaced by an input switch and a direct MFC output is replaced by an output switch.
  A switch node is displayed as a small circle between unit procedures.
- A double-click of a direct MFC input, direct MFC output, input switch, output

switch, confluence, or connector shows the related material parameter(s) within the **Parameter Panel (SR3146.9.2.7)** UI element (page 142).

- The number an MFC input or output shows, corresponds to the numeric suffix of the MFC item identifier, which the system generates and assigns automatically. Before a merge, there are no connections available between inputs and outputs.
- Does not apply to Workflow Designer.
  A confluence node is displayed as circle and represents the point in the material flow when an input is processed and thus converted into an output or merged with other inputs to form an output. A confluence can receive one or more inputs, but can only issue one output.
  The confluence node of an MFC input that has been merged with the final output is displayed as a triangle with an incoming connector and no number.
- Does not apply to Workflow Designer.
  Connectors represent the material flows between the direct MFC input, direct MFC output, confluence nodes, and switch nodes. Connectors between switch nodes are dashed lines if they represent alternative transfers.

Table View (SR3146.9.8.1.1)

The table view of the MFC data provides following features:

- Representation of MFC-related attributes, plus position and planned quantity for ingoing, outgoing, and transfer items
- If the planned quantity mode is **None** or **As produced**, the mode is displayed instead of any planned quantity.
- Synchronization of highlighted MFC items within the table and the graph
- A double-click shows the related material parameter(s) within the **Parameter Panel (SR3146.9.2.7)** UI element (page 142).

Does not apply to Workflow Designer.
The number of a transfer item consists of the numbers of the output and the input items from which it was merged.

Tooltips (SR3146.9.8.2)

Tooltips are available for

- direct MFC inputs,
- direct MFC outputs,
  Does not apply to Workflow Designer.
- confluences, switches, and connectors (transfers).
  Does not apply to Workflow Designer.

They present the following attributes:

- Material identifier
- [Material] Short description,
- Position (if available)
- Planned quantity (if available)
- If the planned quantity mode is **None** or **As produced**, the mode is displayed instead of any planned quantity.
- Path information (not for connectors)

Automatic Merge - Default MFC Definition (SR3146.9.8.3)

➢ Does not apply to Workflow Designer.

The recipe author can trigger the automatic merge of direct MFC inputs, direct MFC outputs, input switches, and output switches at any time during the MFC definition process.

Already merged inputs and outputs are not affected.

Nodes are merged when the system can unambiguously determine their merge targets. When the system identifies two or more merge targets, it does not perform a merge.

DEFAULT MFC DEFINITION WITHIN A UNIT PROCEDURE (SR3146.9.8.4)

➢ Does not apply to Workflow Designer.

The following rules apply to the automatic merge of MFC inputs and outputs within a unit procedure:

- The inputs and outputs of a Dispense unit procedure are merged 1:1, if they have the same material assigned.
- The inputs and outputs of a non-Dispense unit procedure are merged n:1, if only one output is defined. Different materials can be assigned to the inputs and the output.
- In any other case, the inputs and outputs are not merged.
  Already merged inputs and outputs are not affected.

DEFAULT MFC DEFINITION BETWEEN UNIT PROCEDURES (SR3146.9.8.5)

➢ Does not apply to Workflow Designer.

The following rules apply to the automatic merge of MFC inputs and outputs between unit procedures:

- The outputs and inputs are merged 1:1, if there are unique output-input relationships available based on the material assignments and the SFC-related modeling of unit procedures.
  In case of a selection branch (XOR), the merge results in alternative transfers.
- In any other case, the inputs and outputs are not merged.
  Already merged inputs and outputs are not affected.

Manual Merge

MANUAL MERGE WITHIN A UNIT PROCEDURE (SR3146.9.8.6)

➢ Does not apply to Workflow Designer.

The following rules apply to the manual merge of MFC inputs and outputs within a unit procedure:

- Outputs cannot be merged.
- For Dispense unit procedures:
  - Inputs cannot be merged.
  - Inputs and outputs can be merged 1:1, even if they have different materials assigned.
  - When one input (output) is selected, all outputs (inputs) that have the same material assigned are highlighted.

- For non-Dispense unit procedures:
  - Multiple inputs can be merged to one output n:1, even if they have different materials assigned.
  - When one input is selected, all inputs and outputs with which it can potentially be merged are highlighted.
  - When one output is selected, all inputs with which it can potentially be merged are highlighted.

The system can be enabled to support merging multiple inputs to one output not only for non-Dispense, but also for Dispense unit procedures by means of a configuration key (see Technical Manual "Configuration and Extension").

Manual Merge between Unit Procedures (SR3146.9.8.7)

➢ Does not apply to Workflow Designer.

The following rules apply to the manual merge of direct MFC inputs, direct MFC outputs, input switches, and output switches between unit procedures:

- Inputs and outputs can only be merged
  - if they have the same material assigned and
  - if the input belongs to a succeeding unit procedure according to the SFC model or if unit procedures are modeled in parallel and
- In case of simultaneous branches (no XOR), inputs and outputs can only be merged 1:1.
- In case of selection branches (XOR), one input can be merged into multiple alternative outputs and one output can be merged into multiple alternative inputs. In both cases alternative transfers are created.
- When an input (output) is selected, all outputs (inputs) that have the same material assigned and are suitable according to the MFC structure are highlighted.

Manual Merge of MFC Inputs with the Final Output (SR3146.9.8.13)

➢ Does not apply to Workflow Designer.

The following rules apply to the manual merge of MFC inputs with the final output:

- Merge of inputs with the final output is not allowed for Dispense unit procedures (see **Validator - MFC – Final output for Dispense unit procedures (SR3146.9.4.3.118)** message (page 70)).
- Any confluence of a direct MFC input can be merged with the final output of the master recipe.
- Multi-selection is supported.

Split (SR3146.9.8.8)

➢ Does not apply to Workflow Designer.

Inputs and outputs that are presented in a merged status can be split.

Multi-selection of MFC inputs merged with the final output is supported.

- Split connector: dissolves the connection between two nodes.
- Split direct MFC input or output: removes input or output from its current confluence and recreates its initial (single) confluence node.
- Split confluence: dissolves the entire confluence and recreates all initial (single) confluence nodes of its inputs and its outputs.
- Split output/input switch: dissolves all its outgoing/incoming connectors and replaces the switch with its initial output/input.

Reset (SR3146.9.8.9)

➢ Does not apply to Workflow Designer.

The MFC definition can be reset to its initial, unmerged state by dissolving all merges.

Automated Update of MFC Representation (SR3146.9.8.10)

MFC-relevant changes of the master recipe and master workflow result in an automatic update of the graph and the tabular form (e.g. MFC-related material parameters, recipe structure, workflow structure).

MFC data and its representation that is not affected by the change of the master recipe or master workflow remains unchanged.

MFC Definition Linked to Procedure Level (SR3146.9.8.12)

The MFC definition of a master recipe or master workflow is linked to the procedure level, so that MFC definitions can also be modeled as part of a procedure-level custom building block and can be reused along with this custom building block.

Information Flow (SR3146.9.9+)

The information flow supports references of parameters across phases, operations, unit procedures, and supports transitions. The references are maintained in the **Expression editor (SR3146.9.9.4)** UI element (page 148).

Output Mechanism (SR3146.9.9.1)

The output mechanism allows phases to have specific output variables defined.

Referencing (SR3146.9.9.2)

Upon the definition of parameter attributes and transitions, output variables can be referenced.

Only those output variables are available for referencing whose phases can be potentially completed before the execution of the current phase is started, according to the SFC model.

References are supported within an operation and across operations and unit procedures.

No Distributed Execution (SR3146.9.9.2.1)

Output variables that belong to operations marked for Distributed Execution cannot be referenced.

Validator (SR3146.9.9.3)

The automatic validation of references of output variables covers:

- Broken links
- Wrong syntax of expressions/calculations
- Semantic reference errors: referenced phase is not completed when referencing phase becomes active or referenced phase belongs to Dispense operation.

Standard Statuses for Version Control (SR3146.6.4)

The **version control (SR3146.4.5)** function (page 161) supports the following statuses:

- **Initial**: Item is new and has not been saved yet.
- **Edit**: Item is saved and can be modified.
- **Verification**: Item is ready for review. If updates are required, a status change back to **Edit** has to be performed.
- **Scheduled**: Item is valid but its period of validity is in the future related to the current global time.
  This status allows fast filtering for specific items and the creation of a consistent documentation.
- **Valid**: Item is valid. All of its attributes that are under version control cannot be changed.
- **Archived**: Item is no longer valid. It can no longer be used.
- **Obsolete**: Item was never valid and should not be used anymore.

The following important rules apply:

- At a specific time, only one version of an item can be **Valid**.
- To extend the period of validity, a status change from **Valid** to **Valid** must be performed.

Default Version Graph for Master Recipes (SR3146.6.4.1)

The **version control (SR3146.4.5)** function (page 161) supports the following default transitions for master recipes:

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
1.1 - Initial » Edit	---	Automatic transition.
1.2 - Verification » Edit	Vers_Trans_Verification-Edit	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
2.1 - Edit » Verification	Vers_Trans_Edit-Verification	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
3.1 - Verification » Valid	Vers_Trans_Verification-Valid	Default user group: Qualified Person
3.2 - Valid » Valid	Vers_Trans_Valid-Valid	E.g. if period of validity has to be changed. Default user group: Qualified Person
3.3 - Scheduled » Valid	Vers_Trans_Scheduled-Valid	Automatic transition when the current date reaches the start date of the period of validity. Default user group: Qualified Person
4.1 - Valid » Archived	Vers_Trans_Valid-Archived	Automatic transition when the end date of the period of validity is in the past. Automatic transition when the status of another version of a master recipe is set to **Valid** (except if there is a version in the **Scheduled** status). Manual transition when a status transition to **Archived** is executed by the recipe author. Default user group: Qualified Person

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
5.1 - Verification » Scheduled	Vers_Trans_Verification-Scheduled	If the scheduled period overlaps with a period of validity, the end date of the period of validity is moved to a date directly before the start date of the scheduled period. Default user group: Qualified Person
5.2 - Scheduled » Scheduled	Vers_Trans_Scheduled-Scheduled	E.g. if period of validity has to be changed. Default user group: Qualified Person
6.1 - Edit » Obsolete	Vers_Trans_Edit-Obsolete	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
6.2 - Verification » Obsolete	Vers_Trans_Verification-Obsolete	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
6.3 - Scheduled » Obsolete	Vers_Trans_Scheduled-Obsolete	Default user group: Qualified Person

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

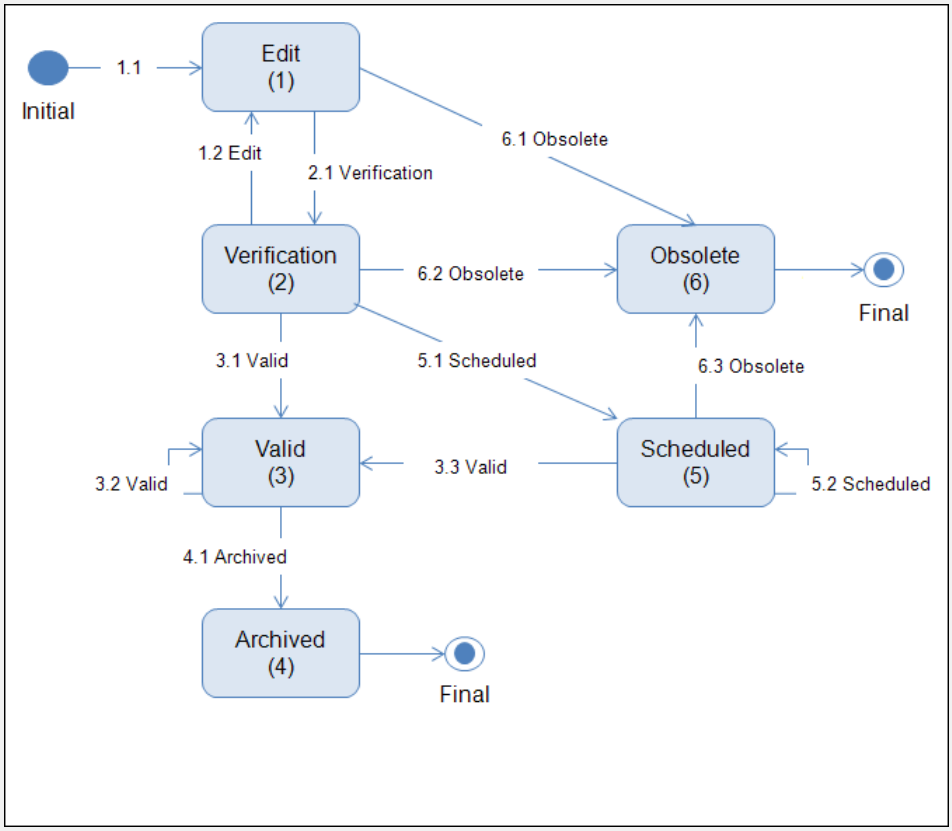


Figure 3: Master recipe - status graph

Default Version Graph for Master Workflows (SR3146.6.4.4)

The **version control (SR3146.4.5)** function (page 161) supports the following default transitions for master workflows:

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
1.1 - Initial » Edit	---	Automatic transition.
1.2 - Verification » Edit	Vers_Trans_WF_Verification-Edit	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
2.1 - Edit » Verification	Vers_Trans_WF_Edit-Verification	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
3.1 - Verification » Valid	Vers_Trans_WF_Verification-Valid	Default user groups: Workflow Author, Qualified Person

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
3.2 - Valid » Valid	Vers_Trans_WF_Valid-Valid	E.g. if period of validity has to be changed. Default user groups: Workflow Author, Qualified Person
3.3 - Scheduled » Valid	Vers_Trans_WF_Scheduled-Valid	Automatic transition when the current date reaches the start date of the period of validity. Default user groups: Workflow Author, Qualified Person
4.1 - Valid » Archived	Vers_Trans_WF_Valid-Archived	Automatic transition when the end date of the period of validity is in the past. Automatic transition when the status of another version of a master workflow is set to **Valid** (except if there is a version in the **Scheduled** status). Manual transition when a status transition to **Archived** is executed by the workflow author. Default user groups: Workflow Author, Qualified Person
5.1 - Verification » Scheduled	Vers_Trans_WF_Verification-Scheduled	If the scheduled period overlaps with a period of validity, the end date of the period of validity is moved to a date directly before the start date of the scheduled period. Default user groups: Workflow Author, Qualified Person
5.2 - Scheduled » Scheduled	Vers_Trans_WF_Scheduled-Scheduled	E.g. if period of validity has to be changed. Default user groups: Workflow Author, Qualified Person
6.1 - Edit » Obsolete	Vers_Trans_WF_Edit-Obsolete	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
6.2 - Verification » Obsolete	Vers_Trans_WF_Verification-Obsolete	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
6.3 - Scheduled » Obsolete	Vers_Trans_WF_Scheduled-Obsolete	Default user groups: Workflow Author, Qualified Person

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

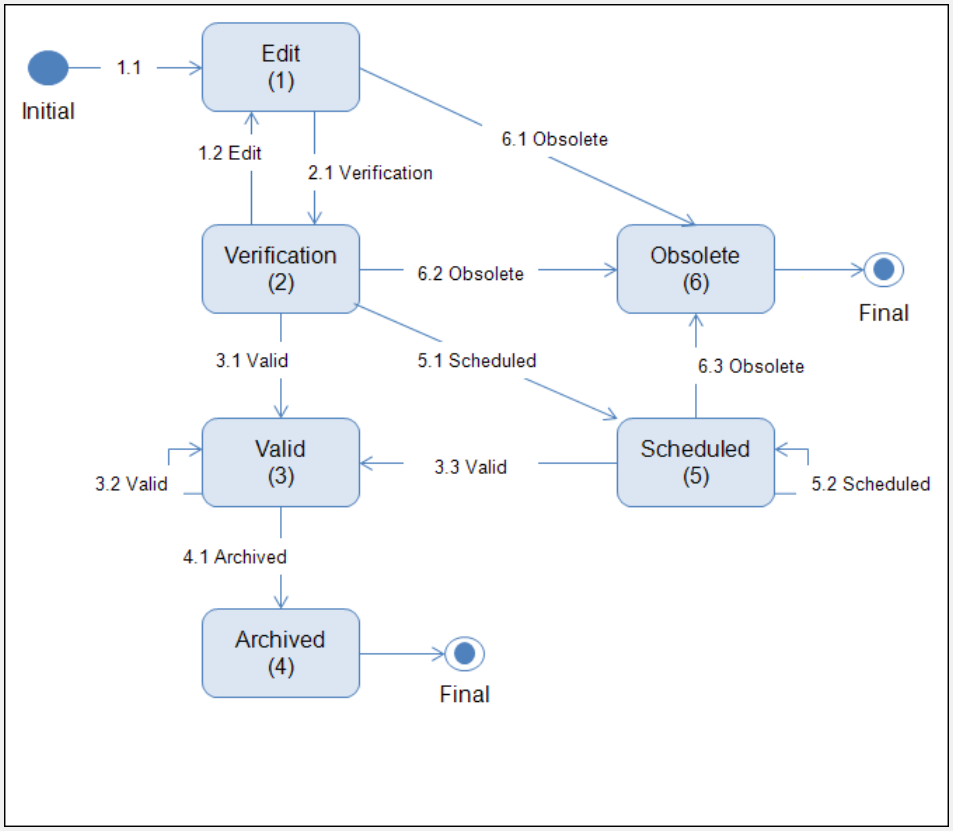


Figure 4: Master workflow - status graph

Validator Consistency Checks (SR3146.9.4.3+)

The following messages are available to inform the recipe or workflow author about issues found by the validator consistency check. They are displayed in the **Messages (SR3146.9.2.16+)** window (page 144).

Checks that Apply to SFC Modeling

Validator - Dummy elements (SR3146.9.4.3.1)

Message category: Error
Issue: Dummy elements are used within a master recipe, master workflow, or (custom) building block.

Validator - Loop (SR3146.9.4.3.3)

Message category: Error
Issue: Loops are built between unit procedures of a master recipe or (custom) building block.

Validator - Outgoing link (SR3146.9.4.3.12)

Message category: Error
Issue: There is no outgoing link defined for a procedural element.

Validator - Start and end step (SR3146.9.4.3.13)

Message category: Error
Issue: Start and end step are not on the same branch.

Validator - Simultaneous branch (SR3146.9.4.3.14)

Message category: Error
Issue: The end of a simultaneous branch is not merged as a simultaneous branch.

Validator - Selection branch (SR3146.9.4.3.15)

Message category: Error
Issue: The end of a selection branch is not merged as a selection branch.

Validator - Simultaneous branch - Unsuitable components (SR3146.9.4.3.17)

Message category: Error
Issue: The end of a simultaneous branch connects unsuitable components (e.g. simultaneous branch includes open ends).

Validator - Errors in preceding components (SR3146.9.4.3.18)

Message category: Warning
Issue: Elements could not be validated due to errors in the preceding components.

Validator - Start step path (SR3146.9.4.3.19)

Message category: Error
Issue: Elements cannot be reached from the start step.

Validator - End step path (SR3146.9.4.3.20)

Message category: Error
Issue: Elements cannot reach the end step.

Validator - Simultaneous branch – End twice (SR3146.9.4.3.148)

Message category: Error
Issue: A simultaneous branch cannot end twice.

Checks that Apply to Group Modeling

Validator - Nested group (SR3146.9.4.3.144)

Message category: Error
Issue: The **Usage as group** attribute of the currently open recipe element is set to **Allowed** or **Mandatory**. Holding a group in its graph is not allowed.

Validator - Graph not allowed for group (SR3146.9.4.3.145+)

The **graph is not v**alid to be used as group. There are several constellations that are not allowed.

Validator – Not allowed branch before end step (SR3146.9.4.3.145.1)

Message category: Error
Issue: An end step of a group must be preceded by a transition, not a branch.

Validator – Not allowed branch after start step (SR3146.9.4.3.145.2)

Message category: Error
Issue: A start step of a group must be succeeded by a transition, not a branch.

Validator – Not allowed branch after start transition (SR3146.9.4.3.145.3)

Message category: Error
Issue: The transition after the start step of a group must be succeeded by an element, not a branch.

Validator – Not allowed branch before end transition (SR3146.9.4.3.145.4)

Message category: Error
Issue: The transition before the end step of a group must be preceded by an element, not a branch.

Validator – Start transition with not allowed condition (SR3146.9.4.3.145.5)

Message category: Error
Issue: The transition after the start step holds a condition, which is not allowed for a group.

Validator – End transition with not allowed condition (SR3146.9.4.3.145.6)

Message category: Error
Issue: The transition before the end step holds a condition, which is not allowed for a group.

Validator – Not allowed parameter for group usage (SR3146.9.4.3.146)

Message category: Error
Issue: The **Usage as group** attribute of the recipe element is set to **Mandatory** and the header component has one or more parameters defined that cannot be retained when it is used as group.

Validator – Not supported parameter for group usage (SR3146.9.4.3.147)

Message category: Information
Issue: The **Usage as group** attribute of the recipe element is set to **Allowed** and the header component has one or more parameters defined that cannot be retained when it is used as group.

Checks that Apply to Transition Conditions

Validator - Transition condition is missing (SR3146.9.4.3.27)

Message category: Error
Issue: There is no condition expression defined for a transition of a selection branch.

Validator - Transition condition - Syntax (SR3146.9.4.3.31)

Message category: Error
Issue: There is a syntax error in the expression of a transition condition.

Validator - Transition condition - Broken link (SR3146.9.4.3.37)

Message category: Depends on the context (see child requirements)
Issue: The expression of a transition condition includes a broken link that needs to be resolved.

Validator - Transition condition - Broken link - Warning (SR3146.9.4.3.37.1)
Context: Custom building block
Message category: Warning (by configuration, the message category can be changed to Error if the transition is flagged for **locking**)

Validator - Transition condition - Broken link - Error (SR3146.9.4.3.37.2)
Context: Master recipe or master workflow
Message category: Error

The broken link can be resolved with the **Messages - Resolve issue (SR3146.9.2.16.4)** function (page 146). The system automatically updates the following data:

- Removes the **XT** marker from the unresolved external reference.

Validator - Transition condition - Invalid reference (SR3146.9.4.3.38)

Message category: Error
Issue: The phase referenced by a transition condition is invalid.

Checks that Apply to Expressions within Parameter Attributes

Validator - Phase parameter - Syntax (SR3146.9.4.3.21)

Message category: Error
Issue: There is a syntax error in the expression of a parameter attribute.

Validator - Phase parameter - Broken link (SR3146.9.4.3.25)

Message category: Depends on the context (see child requirements)
Issue: The expression of a parameter attribute includes a broken link that needs to be resolved.

Validator - Phase parameter - Broken link - Warning (SR3146.9.4.3.25.1)
Context: Custom building block
Message category: Warning (by configuration, the message category can be changed to Error if the parameter is flagged for **locking**)

Validator - Phase parameter - Broken link - Error (SR3146.9.4.3.25.2)
Context: Master recipe or master workflow
Message category: Error

The broken link can be resolved with the **Messages - Resolve issue (SR3146.9.2.16.4)** function (page 146). The system automatically updates the following data:

- Removes the **XT** marker from the unresolved external reference.

Validator - Phase parameter - Invalid reference (SR3146.9.4.3.36)

Message category: Error
Issue: The phase referenced by a parameter attribute is invalid.

Checks that Apply to MFC Modeling of a Procedure within Recipe and Workflow Designer

Validator - MFC - Final output does not match material (product) (SR3146.9.4.3.61)

➢ Does not apply to Workflow Designer.

Message category: Error
Issue: The final output material of an MFC definition does not match the material (product) defined within the master recipe header.

Validator - MFC - Multiple outputs for non-Dispense unit procedures (SR3146.9.4.3.62)

➢ Does not apply to Workflow Designer.

Message category: Depends on the context (see child requirements)
Issue: For non-standalone Dispense unit procedures, there are multiple direct MFC outputs defined or the one defined output is not the final output of the recipe.

Validator - MFC - Multiple outputs for non-Dispense unit procedures - Warning (SR3146.9.4.3.62.1)
Context: Custom building block
Message category: Warning

Validator - MFC - Multiple outputs for non-Dispense unit procedures - Error (SR3146.9.4.3.62.2)
Context: Master recipe
Message category: Error

Validator - MFC - No output defined for input (SR3146.9.4.3.63)

- Does not apply to Workflow Designer.

Message category: Depends on the context (see child requirements)
Issue: There is no target output defined for a direct MFC input or an input switch.

Validator - MFC - No output defined for input - Warning (SR3146.9.4.3.63.1)
Context: Custom building block
Message category: Warning

Validator - MFC - No output defined for input - Error (SR3146.9.4.3.63.2)
Context: Master recipe
Message category: Error

Validator - MFC - No input defined for output (SR3146.9.4.3.64)

- Does not apply to Workflow Designer.

Message category: Depends on the context (see child requirements)
Issue: There is no predecessor input defined for a direct MFC output or an output switch.

Validator - MFC - No input defined for output - Warning (SR3146.9.4.3.64.1)
Context: Custom building block
Message category: Warning

Validator - MFC - No input defined for output - Error (SR3146.9.4.3.64.2)
Context: Master recipe
Message category: Error

Validator - MFC - Output defined for workflow (SR3146.9.4.3.110)

- Does not apply to Recipe Designer.

Message category: Error
Issue: An output has been defined for a master workflow.

Validator - MFC - No direct output defined (SR3146.9.4.3.117)

- Does not apply to Workflow Designer.

Message category: Depends on the context (see child requirements)
Issue: There is no direct output defined or at least, for one execution path, no direct output is defined.

Validator - MFC - No direct output defined - Warning (SR3146.9.4.3.117.1)
Context: Custom building block
Message category: Warning

Validator - MFC - No direct output defined - Error (SR3146.9.4.3.117.2)
Context: Master recipe
Message category: Error

Validator - MFC - Attributes of material parameters do not match (SR3146.9.4.3.136)

- Does not apply to Workflow Designer.

Message category: Error
Issue: The attributes of material parameters with the same MFC position are different.

Validator - MFC - Outgoing or incoming connector is missing (SR3146.9.4.3.138)

- Does not apply to Workflow Designer.

Message category: Error
Issue: For at least one execution path, there is no outgoing connector defined for a switch output or no incoming connector defined for a switch input.

Validator - MFC - More than one outgoing or incoming connector on one execution path (SR3146.9.4.3.139)

- Does not apply to Workflow Designer.

Message category: Error
Issue: For at least one execution path, a switch output has more than one outgoing connector or a switch input has more than one incoming connector.

Checks that Apply to Master Recipes in General

Validator - Recipe - Reference quantity not defined (SR3146.9.4.3.67)

- Does not apply to Workflow Designer.

Message category: Error
Issue: The planned quantity of a master recipe or the UoM of the planned quantity is missing.
Per default configuration, the check is disabled for master recipes of the **Cost center** usage type.

Validator - Recipe - Min/Max quantities without UoM (SR3146.9.4.3.111)

- Does not apply to Workflow Designer.

Message category: Error
Issue: The minimum or maximum quantity-related unit of measure of a master recipe is missing or the unit of measure cannot be converted with the available conversion factors.

Validator - Master recipe and material parameter - Conflicting inventory level (SR3146.9.4.3.52)

Message category: Error
Issue: There are conflicts in the inventory level definition of the packaging levels of the master recipe and its material output parameters. Applies only to material output parameters that refer to the material (product) of the master recipe.

Validator - Packaging level of master recipe (SR3146.9.4.3.107)

Message category: Error
Issue: For the packaging level defined in the master recipe header there is a contained number defined but no meaning.

Checks that Apply to Master Recipes Based on an ERP BOM

Validator - Recipe - Planned quantity does not fit ERP BOM (SR3146.9.4.3.57)

➢ Does not apply to Workflow Designer.

Message category: Error
Issue: The planned quantity of the master recipe does not match the base quantity of the related ERP BOM.

Validator - Recipe - Planned quantity does not fit ERP BOM - Resolve (SR3146.9.4.3.57.1)
The inconsistency can be resolved with the **Messages - Resolve issue (SR3146.9.2.16.4)** function (page 146). The system automatically updates the following data:

- Planned quantity of the master recipe

The automatic resolution of the ERP BOM discrepancy is also included in the **Workbench Object Operations - Replace Material (Product) (SR3146.9.10.11)** operation (page 42).

Validator - Recipe - Packaging level does not fit ERP BOM (SR3146.9.4.3.58)

➢ Does not apply to Workflow Designer.

Message category: Error
Issue: The packaging level (content) of the master recipe does not match the packaging level (content) of the related ERP BOM.

Validator - Recipe - Packaging level does not fit ERP BOM - Resolve (SR3146.9.4.3.58.1)
The inconsistencies can be resolved with the **Messages - Resolve issue (SR3146.9.2.16.4)** function (page 146). The system automatically updates the following data:

- Packaging level of the master recipe
- Packaging level content of the master recipe

The automatic resolution of the ERP BOM discrepancies is also included in the **Workbench Object Operations - Replace Material (Product) (SR3146.9.10.11)** operation (page 42).

Validator - Phase parameter - Missing ERP BOM position (SR3146.9.4.3.50)

- Does not apply to Workflow Designer.

Message category: Error
Issue: In a master recipe that is based on an ERP BOM, there are positions that represent a direct MFC input missing in the material parameters.

Validator - Phase parameter - Multiple ERP BOM positions (SR3146.9.4.3.51)

- Does not apply to Workflow Designer.

Message category: Error
Issue: There are positions defined multiple times in the material parameters of a master recipe or procedure building block that is based on an ERP BOM. The positions are not unique throughout an entire execution path.

Validator - Phase parameter - Non-ERP BOM position (SR3146.9.4.3.55)

- Does not apply to Workflow Designer.

Message category: Error
Issue: In a master recipe that is based on an ERP BOM, there are material parameter positions that represent direct MFC inputs, which are not part of the ERP BOM.

Validator - Phase parameter - ERP BOM attribute discrepancy (SR3146.9.4.3.56)

- Does not apply to Workflow Designer.

Message category: Error
Issue: At least one of the common attributes of a position of a material parameter and the master recipe-related ERP BOM item do not match.
Common attributes are material identifier and short description, quantity with UoM, Fixed quantity.

Validator - Phase parameter - ERP BOM attribute discrepancy - Resolve (SR3146.9.4.3.56.1)
The inconsistencies can be resolved with the **Messages - Resolve issue (SR3146.9.2.16.4)** function (page 146). The system automatically updates the following data:

- Material inputs by BOM position (material identifier, material short description, planned quantity, and fixed quantity)
- In case linked MFC inputs/outputs, according to the MFC definition, refer to the same material:
  The system also updates the material identifier and planned quantity of the related material outputs (same unit procedure, e.g. Dispense, planned quantity only for the first linked output) and the material identifier of the subsequent material inputs (subsequent unit procedure, e.g. Mixing).

The automatic resolution of the ERP BOM discrepancies is also included in the **Workbench Object Operations - Replace Material (Product) (SR3146.9.10.11)** operation (page 42).

Validator - MFC (final output) - Planned quantity does not fit master recipe (SR3146.9.4.3.133)

➢ Does not apply to Workflow Designer.

Message category: Information
Issue: The planned quantity of the final output does not match the planned quantity to be produced by the master recipe.

Checks that Apply to Material Parameters of a Phase

Validator - Minimum required material parameters (SR3146.9.4.3.10)

Message category: Depends on the context (see child requirements)
Issue: The number of minimum required material input or output parameters of a phase is not met.

Validator - Minimum required material parameters - Warning (SR3146.9.4.3.10.1)
Context: Custom building block
Message category: Warning

Validator - Minimum required material parameters - Error (SR3146.9.4.3.10.2)

Context: Master recipe and master workflow
Message category: Error
Per default configuration, the check is disabled for master recipes of the **Cost center** usage type.

Validator - MFC - Input without planned quantity (SR3146.9.4.3.66)

Message category: Depends on the context (see child requirements)
Issue: The planned quantity of an MFC-related input parameter that represents a direct input is missing.

Validator - MFC - Input without planned quantity - Warning (SR3146.9.4.3.66.1)
Context: Custom building block, on procedure level only
Message category: Warning

Validator - MFC - Input without planned quantity - Error (SR3146.9.4.3.66.2)
Context: Master recipe and master workflow
Message category: Error

Validator - MFC - Uniqueness and format of MFC position (SR3146.9.4.3.68)

Message category: Error
Issue:

- The MFC positions assigned to MFC-related parameters (input and output) are not unique throughout an entire execution path or
- for filler materials, the assigned MFC position is not numeric.
  The filler-related check only applies in the context of batch-related recipes and custom building blocks on procedure level.

Validator - Material - Incompatible UoM (SR3146.9.4.3.69)

Message category: Error
Issue: The UoM assigned to an attribute of a material parameter is not supported or incompatible with the UoM of the material. The following attributes are affected:

- Planned quantity, Target weight
- Non-weighable UoM in case of a Dispense operation
  Does not apply to Workflow Designer.
- UoM cannot be converted in case of an input or output material parameter of a Dispense operation
  Does not apply to Workflow Designer.
- Absolute tolerances (of planned quantity and target weight)
- Relative tolerances (of planned quantity and target weight)

Validator - Material - UoM of planned potency (SR3146.9.4.3.70)

Message category: Error
Issue: The UoM of the planned potency of a material parameter is not suitable.

Validator - Material - One tolerance is missing (SR3146.9.4.3.71)

Message category: Error
Issue: Only one tolerance of a material parameter is defined and either the lower or the upper tolerance is missing. If a target weight has been defined, either its lower or upper tolerance is missing.

Validator - Material - Negative values (SR3146.9.4.3.77)

Message category: Error
Issue: The defined quantity or tolerance for a material parameter attribute is negative. If a target weight has been defined, its quantity or tolerance is negative.

Validator - MFC - MFC position is missing (SR3146.9.4.3.78)

Message category: Depends on the context (see child requirements)
Issue: There is no MFC position assigned to an MFC-related input parameter that represents a direct input.

Validator - MFC - MFC position is missing - Warning (SR3146.9.4.3.78.1)
Context: Custom building block, on procedure level only
Message category: Warning

Validator - MFC - MFC position is missing - Error (SR3146.9.4.3.78.2)
Context: Master recipe and master workflow
Message category: Error

Validator - BOM position not used (SR3146.9.4.3.137)

Message category: Error
Issue: For a particular BOM position, there is no corresponding direct MFC input on at least one execution path.

Validator - MFC - Unallowed planned quantity mode (SR3146.9.4.3.113)

Message category: Depends on the context (see child requirements)
Issue: The planned quantity mode setting of an MFC-relevant material parameter is not allowed.

- The **As produced** planned quantity mode is only allowed for ingoing MFC transfer items.
- The **None** planned quantity mode is only allowed for materials of the Auxiliary substance weighing material type (or if no weighing material type is set).

Validator - MFC - Unallowed planned quantity mode - Warning (SR3146.9.4.3.113.1)
Context for the **As produced** planned quantity mode: Custom building block, on procedure level only
Message category: Warning

Validator - MFC - Unallowed planned quantity mode - Error (SR3146.9.4.3.113.2)
Context for the **As produced** planned quantity mode: Master recipe and master workflow
Context for the **None** planned quantity mode: Custom building block, master recipe, and master workflow
Message category: Error

Validator - MFC - Fixed quantity not allowed (SR3146.9.4.3.114)

Message category: Information
Issue: An MFC-relevant output parameter is set to have a fixed quantity.

Validator - MFC - Fixed quantity not allowed for Active or Compensator (SR3146.9.4.3.128)

Message category: Information
Issue: An MFC-relevant material input parameter whose weighing material type is **Active** or **Compensator** is set to have a fixed quantity.

Validator - MFC - Planned quantities of input and output parameters do not match (SR3146.9.4.3.116)

Message category: Information
Issue: The planned quantity of an MFC-relevant input parameter does not match the planned quantity of its output parameter.

Validator - Material parameter - Conflicting inventory level (SR3146.9.4.3.53)

Message category: Error
Issue: There are conflicts in the inventory level definition of the packaging levels. Applies only to material parameters that do not refer to the material (product) of the master recipe.

Validator - Packaging level of material parameter (SR3146.9.4.3.108)

Message category: Error
Issue: A packaging level of a material parameter has a contained number defined but no meaning.

Validator - Material parameter - Packaging level does not fit material's master data (SR3146.9.4.3.109)

Message category: Error
Issue: The packaging level (content) of the material parameter does not match the packaging level (content) of the material's master data.

Validator - Material parameter - Packaging level does not fit material's master data - Resolve (SR3146.9.4.3.109.1)
The inconsistencies can be resolved with the **Messages - Resolve issue (SR3146.9.2.16.4)** function (page 146). The system automatically updates the following data:

- Packaging level of the material parameter
- Packaging level content of the material parameter

The automatic resolution of the ERP BOM discrepancies is also included in the **Workbench Object Operations - Replace Material (Product) (SR3146.9.10.11)** operation (page 42).

Checks that Apply to Specific Dispense Building Blocks

Validator - Dispense operation (SR3146.9.4.3.35)

Message category: Error
Issue: The unit procedure of a Dispense operation holds more than one operation.

Validator - MFC - Transfer input for Dispense unit procedures (SR3146.9.4.3.65)

➢ Does not apply to Workflow Designer.

Message category: Depends on the context (see child requirements)
Issue: An MFC transfer is defined as an input material for a Dispense unit procedure.

Validator - MFC - Transfer input for Dispense unit procedures - Warning (SR3146.9.4.3.65.1)
Context: Custom building block
Message category: Warning

Validator - MFC - Transfer input for Dispense unit procedures - Error (SR3146.9.4.3.65.2)
Context: Master recipe
Message category: Error

Validator - MFC - Final output for Dispense unit procedures (SR3146.9.4.3.118)

- Does not apply to Workflow Designer.

Message category: Depends on the context (see child requirements)
Issue: One of the materials of a Dispense unit procedure is merged with the final output.

Validator - MFC - Final output for Dispense unit procedures - Warning (SR3146.9.4.3.118.1)
Context: Custom building block
Message category: Warning

Validator - MFC - Final output for Dispense unit procedures - Error (SR3146.9.4.3.118.2)
Context: Master recipe
Message category: Error

Validator - MFC - None planned quantity mode for Dispense unit procedure (SR3146.9.4.3.115)

Message category: Error
Issue: An MFC-relevant material parameter with **None** as planned quantity mode is defined for a Dispense unit procedure.

Validator - D_Material - Default weighing method not defined (SR3146.9.4.3.72)

- Does not apply to Workflow Designer.

Message category: Error
Issue: The default weighing method is not defined for a material parameter of a Dispense operation.

Validator - D_Material - Default weighing method not allowed (SR3146.9.4.3.132)

- Does not apply to Workflow Designer.

Message category: Error
Issue: The default weighing method of a material input parameter of a Dispense operation is not among its allowed weighing methods.

Validator - D_Material - Allowed weighing methods not defined (SR3146.9.4.3.73)

- Does not apply to Workflow Designer.

Message category: Error
Issue: The allowed weighing methods are not defined for a material parameter of a Dispense operation.

Validator - D_Material - Tolerances not defined (SR3146.9.4.3.74)

- Does not apply to Workflow Designer.

Message category: Error
Issue:

- Dispense: The tolerances are not defined for a material parameter of a Dispense operation.
- Output weighing: If a target weight has been defined, its tolerances are not defined.

Validator - D_Material - Weighing material type not defined (SR3146.9.4.3.75)

- Does not apply to Workflow Designer.

Message category: Error
Issue: The weighing material type is not defined for a material parameter of a Dispense operation.

Validator - D_Material - Planned potency not defined (SR3146.9.4.3.76)

- Does not apply to Workflow Designer.

Message category: Error
Issue: The planned potency is not defined for an active material of a Dispense operation.

Validator - D_Material - Weighing sequence not defined (SR3146.9.4.3.141)

- Does not apply to Workflow Designer.

Message category: Error
Issue: A weighing sequence is defined for at least one material input parameter of the **Identify material** phase of a Dispense operation, but not for all MFC-relevant material input parameters of the phase.

Validator - D_Material - Weighing sequence and wrong usage type (SR3146.9.4.3.142)

- Does not apply to Workflow Designer.

Message category: Error
Issue: A weighing sequence is defined for a material input parameter of the **Identify material** phase of a Dispense operation, but this is only supported for the **Production** usage type.

Checks that Apply to Inline Weighing

Validator - MFC - Unallowed weighing method (SR3146.9.4.3.120)

Message category: Error
Issue: For the **Identify material** phase of an Inline Weighing operation, **Pallet** weighing is not allowed for MFC-relevant material parameters.

Validator - MFC - Planned quantity mode is As defined - Tolerances are missing (SR3146.9.4.3.121)

Message category: Error
Issue: For the **Identify material** phase of an Inline Weighing operation, there are no tolerances defined for an MFC-relevant material parameter with **As defined** as planned quantity mode.

Validator - MFC - Planned quantity mode is As defined - Planned quantity is missing (SR3146.9.4.3.122)

Message category: Error
Issue: For the **Identify material** phase of an Inline Weighing operation, there is no planned quantity defined for an MFC-relevant material parameter with **As defined** as planned quantity mode.

Validator - MFC - Weighing material type not defined (SR3146.9.4.3.123)

Message category: Error
Issue: For the **Identify material** phase of an Inline Weighing operation, there is no weighing material type defined for an MFC-relevant material parameter.

Validator - MFC - MFC position is missing (SR3146.9.4.3.124)

Message category: Error
Issue: For the **Identify material** phase of an Inline Weighing operation, there is no MFC position assigned to an MFC-related input material parameter. This applies to ingoing MFC transfer item material parameters only.

Validator - MFC - Ingoing MFC transfer item - Unallowed weighing material type (Active) (SR3146.9.4.3.125)

Message category: Error
Issue: For the **Identify material** phase of an Inline Weighing operation, the **Active** weighing material type is not allowed for an ingoing MFC transfer item.

See also **Unallowed weighing material type (Compensator, Filler) (SR3146.9.4.3.134)** message (page 73).

Validator - MFC - Ingoing MFC transfer item - Unallowed weighing material type (Compensator, Filler) (SR3146.9.4.3.134)

Message category: Error
Issue: For the **Identify material** phase of an Inline Weighing operation, the **Compensator** and **Filler** weighing material types are not allowed for an ingoing MFC transfer item if they were already set to a weighing material type other than **Auxiliary** before in at least one preceding material position.

See also **Unallowed weighing material type (Active) (SR3146.9.4.3.125)** message (page 72).

Validator - MFC - Ingoing MFC transfer item - Unallowed planned quantity mode (SR3146.9.4.3.135)

Message category: Error
Issue: For the **Identify material** phase of an Inline Weighing operation, the **As produced** planned quantity mode is not allowed for an ingoing MFC transfer item of the **Compensator** and **Filler** weighing material types.

Validator - D_Material - Planned potency not defined (SR3146.9.4.3.127)

- Does not apply to Workflow Designer.

Message category: Error
Issue: The planned potency is not defined for an active material of an Inline Weighing operation.

Validator - MFC - Planned quantity mode is As produced - Tolerances are missing (SR3146.9.4.3.119)

Message category: Error
Issue: In an Inline Weighing operation, there are no tolerances defined for an MFC-relevant parameter with **As produced** as planned quantity mode.

Validator - D_Material - Weighing sequence not defined (SR3146.9.4.3.143)

- Does not apply to Workflow Designer.

Message category: Error
Issue: A weighing sequence is defined for at least one material input parameter of the **Identify material** phase of an Inline Weighing operation, but not for all MFC-relevant material input parameters of the phase.

Checks that Apply to Output Weighing

Validator - MFC - Planned quantity mode is As defined - Tolerances are missing (SR3146.9.4.3.129)

Message category: Error
Issue: For the **Weigh** phase of an Output Weighing operation, there are no tolerances defined for an MFC-relevant material parameter with **As defined** as planned quantity mode.

Validator - MFC - Planned quantity mode is As defined - Planned quantity is missing (SR3146.9.4.3.130)

Message category: Error
Issue: For the **Weigh** phase of an Output Weighing operation, there is no planned quantity defined for an MFC-relevant material parameter with **As defined** as planned quantity mode.

Validator - MFC - Planned quantity mode is As defined - Allowed weighing method is missing (SR3146.9.4.3.131)

Message category: Error
Issue: For the **Weigh** phase of an Output Weighing operation, there is no allowed weighing method defined for an MFC-relevant material parameter with **As defined** as planned quantity mode.

Additional checks for Output Weighing

- Validator - D_Material - Tolerances Not Defined (SR3146.9.4.3.74) check (page 71)

Checks that Apply to Non-production Master Recipes in General

Validator - Non-production MR - More than one unit procedure or operation (SR3146.9.4.3.100)

Message category: Error
Issue: The procedural structure of a non-production master recipe contains more than one unit procedure or operation.

Checks that are disabled for master recipes of the Cost center usage type

- Validator - Minimum required material parameters - Error (SR3146.9.4.3.10.2) check (page 65)
- Validator - Recipe - Reference quantity not defined (SR3146.9.4.3.67) check (page 62)

Checks that Apply to Master Workflows in General

Validator - Workflow - Phase is incompatible (SR3146.9.4.3.80)

Message category: Error
Issue: The inserted custom building block contains phases that are not compatible with master workflows.

Validator - Workflow - Workflow type not defined (SR3146.9.4.3.81)

Message category: Error
Issue: The workflow type is missing that defines to which group the one-click startable workflow belongs in the Cockpit of the Production Execution Client.

Validator - Workflow - Only appendable if production-relevant (SR3146.9.4.3.112)

Message category: Error
Issue: A master workflow is not production-relevant, but is set to be appendable during processing.

Validator - Workflow - Processing name not defined (SR3146.9.4.3.54)

Message category: Error
Issue: The processing name that is displayed during execution of the workflow is missing.

Checks that Apply to Signature Privileges of a Phase

Validator - Usage type (SR3146.9.4.3.8)

Message category: Error
Issue: The same usage type is defined for multiple signature privileges of a single phase.

Validator - Exception signature supported (SR3146.9.4.3.9)

Message category: Error
Issue: An exception signature is defined, although exceptions are not supported by the phase.

Validator - Double signature for sequential phase completion (SR3146.9.4.3.5)

Message category: Error
Issue: A double signature for a sequential phase completion signature has been defined.

Validator - (Sequential) phase completion signatures (SR3146.9.4.3.6)

Message category: Error
Issue: One phase completion signature and one sequential phase completion signature have been assigned to a phase.

Checks that Apply to Equipment Requirements of a Phase

Validator - Property definition is inconsistent (SR3146.9.4.3.82)

Message category: Error
Issue: The content definition of a property type does not match the content definition of the respective class property.

Validator - Minimum required equipment parameters (SR3146.9.4.3.83)

Message category: Depends on the context (see child requirements)
Issue: The number of minimum required equipment requirement parameters of a phase is not met.

Validator - Minimum required equipment parameters - Warning (SR3146.9.4.3.83.1)
Context: Custom building block
Message category: Warning

Validator - Minimum required equipment parameters - Error (SR3146.9.4.3.83.2)
Context: Master recipe and master workflow
Message category: Error

Validator - Equipment class requirements are not unique (SR3146.9.4.3.85)

Message category: Error
Issue: The equipment requirements on class level are not unique. One equipment class has been assigned multiple times.

Validator - Equipment requirement - BOE is not unique (SR3146.9.4.3.88)

Message category: Error
Issue: The BOE position assigned to an equipment requirement parameter is not unique throughout the master recipe, master workflow, or building block.

Validator - Equipment requirement identifier is not unique (SR3146.9.4.3.86)

Message category: Error
Issue: The equipment requirement identifier is not unique within a class.

Validator - Equipment requirement rule - Identifier is missing (SR3146.9.4.3.87)

Message category: Error
Issue: There is no identifier defined for an equipment requirement rule.

Validator - Equipment requirement rule - Syntax error (SR3146.9.4.3.89)

Message category: Error
Issue: There is a syntax error in the expression of an equipment requirement rule.

Validator - Equipment requirement rule - Broken link (SR3146.9.4.3.90)

Message category: Depends on the context (see child requirements)
Issue: The expression of an equipment requirement rule includes a broken link that needs to be resolved.

Validator - Equipment requirement rule - Broken link - Warning (SR3146.9.4.3.90.1)
Context: Custom building block
Message category: Warning (by configuration, the message category can be changed to Error if the parameter is flagged for **locking**)

Validator - Equipment requirement rule Broken link - Error (SR3146.9.4.3.90.2)
Context: Master recipe or master workflow
Message category: Error

The broken link can be resolved with the **Messages - Resolve issue (SR3146.9.2.16.4)** function (page 146). The system automatically updates the following data:

- Removes the **XT** marker from the unresolved external reference.

Validator - Equipment requirement rule - Invalid reference (SR3146.9.4.3.91)

Message category: Error
Issue: The phase referenced by an equipment requirement rule is invalid.

Validator - Equipment requirement rule - Rule expression is missing (SR3146.9.4.3.92)

Message category: Error
Issue: There is no rule expression defined for an equipment requirement rule.

Checks that Apply to Work Center Assignments

Station assignment does not match a given work center assignment (SR3146.9.4.3.40)

Message category: Error
Issue: Per unit procedure, at least one station assigned on operation level does not match the work centers assigned on unit procedure level. Therefore, work centers have to be added or stations have to be removed.

Station assignment is missing for a given work center assignment (SR3146.9.4.3.42)

Message category: Error
Issue: At least one station is already assigned to an operation. However, there are more work centers assigned on unit procedure level and the dispatching of this operation does not yet reflect the stations of all of the work centers.

Unusable station (SR3146.9.4.3.43)

Message category: Error
Issue: At least one station of this operation cannot be used because it does not belong to any work center.

Unusable work center (SR3146.9.4.3.44)

Message category: Error
Issue: At least one work center of this unit procedure cannot be used because it has no station defined.

Checks that Apply to Capability Assignments in General

Usage of capability not allowed for workflows (SR3146.9.4.3.79)

Message category: Error
Issue: The operation/unit procedure contains the Abort-and-reactivate-enabled capability. But the capability is not permitted for use in workflows.

Usage of capability not allowed for Dispense (SR3146.9.4.3.45)

Message category: Error
Issue: The operation/unit procedure contains at least one of the following capabilities: Abort-and-reactivate-enabled, Auto-startable, Detachable, Escalation-enabled, Event-triggered, Server-run, or Trigger-enabled. These capabilities, however, are not permitted for use in Dispense operations or unit procedures.

Usage of capability not allowed twice for simultaneous branches (SR3146.9.4.3.106)

Message category: Error
Issue: There is more than one operation that holds the Auto-startable capability, which is not allowed for operations located on a simultaneous branch.

Capability assignment does not match usage type of privilege parameter (SR3146.9.4.3.93)

Message category: Depends on the context (see child requirements)
Issue: The operation or unit procedure contains a capability, but has no signature privilege of the same usage type to match the capability's risk assessment.

Capability assignment does not match usage type of privilege parameter - Warning (SR3146.9.4.3.93.1)
Context: Custom building block
Message category: Warning

Capability assignment does not match usage type of privilege parameter - Error (SR3146.9.4.3.93.2)
Context: Master recipe and master workflow
Message category: Error

Incompatible capability assignment (SR3146.9.4.3.46)

Message category: Error
Issue: Capabilities that are assigned to the same operation are not compatible. This applies to the following combinations:

Capability	Auto-startable	Server-run
Auto-startable	N/A	Incompatible*
Detachable	Compatible	Incompatible
Escalation-enabled	Incompatible	Incompatible
Event-triggered	Incompatible	Incompatible
Trigger-enabled	Compatible	Incompatible

*Server-run operations will start automatically by design. The Auto-startable capability only applies as an option to operations that run on a client.

Incompatible capability assignment between different levels (SR3146.9.4.3.105)

Message category: Error
Issue: Capabilities that are assigned to a unit procedure are not compatible with the capabilities of its operations. This applies to the following combinations:

Capability	Detachable (unit procedure)
Server-run (operation)	Incompatible
Event-triggered (operation)	Incompatible

Capability assignment is missing (SR3146.9.4.3.47)

Message category: Error
Issue: A capability has been assigned, which also requires another capability. This applies to the Trigger-enabled and Escalation-enabled capabilities, which are only allowed in combination with the Event-triggered capability.

Checks that Apply to Escalation-enabled Capability Assignments

Escalation validator - Duration is missing for enabled exceptions (SR3146.9.4.3.94)

Message category: Error
Issue: The Escalation-enabled capability has been assigned and exceptions are enabled, but not all related durations are defined.

Escalation validator - Durations are conflicting (SR3146.9.4.3.95)

Message category: Error
Issue: The Escalation-enabled capability has been assigned, but the durations defined for its actions are conflicting.

Escalation validator - No duration defined (SR3146.9.4.3.96)

Message category: Depends on the context (see child requirements)
Issue: The Escalation-enabled capability has been assigned, but no duration is defined.

Escalation validator - No duration defined - Warning (SR3146.9.4.3.96.1)
Context: Custom building block
Message category: Warning

Escalation validator - No duration defined - Error (SR3146.9.4.3.96.2)
Context: Master recipe and master workflow
Message category: Error

Checks that Apply to Trigger-enabled Capability Assignments

Trigger validator - No valid trigger reference defined (SR3146.9.4.3.97)

Message category: Depends on the context (see child requirements)
Issue: A Trigger-enabled capability contains triggers with a missing or invalid trigger phase reference.

Trigger validator - No valid trigger reference defined - Warning (SR3146.9.4.3.97.1)
Context: Custom building block
Message category: Warning

Trigger validator - No valid trigger reference defined - Error (SR3146.9.4.3.97.2)
Context: Master recipe and master workflow
Message category: Error

Trigger validator - Trigger identifiers are not unique (SR3146.9.4.3.98)

Message category: Error
Issue: A Trigger-enabled capability contains triggers whose identifiers are not unique.

Trigger validator - Triggers are missing (SR3146.9.4.3.99)

Message category: Error
Issue: A Trigger-enabled capability does not contain any triggers.

Trigger phase without event-triggered operation (SR3146.9.4.3.101)

Message category: Information
Issue: A trigger phase is used in the recipe or workflow that is not referenced from an event-triggered operation.

Checks that Apply to Server-run Capability Assignments

Server-run validator - Conflicting station assignment (SR3146.9.4.3.102)

Message category: Error
Issue: The Server-run capability has been assigned, but stations are also assigned.

Server-run validator - Phase cannot be run on a server (SR3146.9.4.3.103)

Message category: Error
Issue: The Server-run capability has been assigned, but the phase cannot be run on a server.

Server-run validator - Phase needs to be run on a server (SR3146.9.4.3.104)

Message category: Error
Issue: The Server-run capability has not been assigned, but the phase needs to be run on a server.

Server-run validator - No phase completion signature or mandatory comment allowed (SR3146.9.4.3.7)

Message category: Error
Issue: A (sequential) phase completion signature or a signature with a mandatory comment has been assigned to a phase that needs to be run on a server.

Checks that Apply to the Wait for Event (OES) Phase

Validator - Send event phase is missing (SR3146.9.4.3.140)

Message category: Warning
Issue: For a unit procedure that holds a **Wait for event (OES)** phase, there is no **Send event** phase available within the unit procedure.

Chapter 4

Management of Building Blocks (SR3146.10+)

The management of building blocks differs from the management of master recipes and master workflows in the following areas:

Header Attributes (SR3146.10.4)

The custom building block object supports the following header attributes:

- Basic data:
  - Identifier
  - Revision
  - Short description
  - Comparison baseline
  - Usage as <operation or phase> group
    Only for operations and unit procedures.
  - Access privilege
  - Category 1 - 2
    To display Category 3, the system needs to be configured. For details, see "Configuring the Parameter Panel of Recipe and Workflow Designer" in Volume 3 of the "Technical Manual Configuration and Extension" [A5] (page 171).
  - Description
  - Dispense
    Only for operations. Read-only for unit procedures.
  - Operator exception texts
    Only for phases, operations, and unit procedures.
  - Reviewer exception texts
    Only for phases, operations, and unit procedures.
  - Hide in batch/workflow report
    Only for phases.

- Status data (for the full history, see the **Status History (SR3146.10.5)** function (page 88)):
  - Status
  - Approved on
  - Approved by
  - Archived on
  - Archived by

The attributes are maintained in the **header component-specific (SR3146.9.2.15)** property window (page 144).

Workbench Object Operations (SR3146.9.10.7)

The following operations are supported for custom building blocks. They apply to elements on all levels (procedure, unit procedure, operation, phase).

- Create new building block (page 86)
- Open existing (see **Open (SR3146.9.2.20+)** dialog (page 136))
- View locked building block (page 86)
- Save (current or all)
- Save as new custom BB
  For custom building blocks that are protected by an access privilege: The access privilege of the original custom building block is passed on to the new custom building block. In the default configuration, the access privilege cannot be changed.
- Rename
- Copy selected element(s), see **Copy & Paste Function (SR3146.9.1.6+)** feature (page 133)
- Paste previously copied element(s), see **Copy & Paste Function (SR3146.9.1.6+)** feature (page 133)
- Create <same BB level> from selected element
  For custom building blocks that are protected by an access privilege: The access privilege of the original custom building block is passed on to the new custom building block. In the default configuration, the access privilege cannot be changed.

- Create <next higher BB level> from selected element(s)
  For custom building blocks that are protected by an access privilege: The access privilege of the original custom building block is passed on to the new custom building block. In the default configuration, the access privilege cannot be changed.
- Replace (overrides parameters) via the **Setlist (SR3146.9.2.9)** UI element (page 140)
- Smart replace via the **Setlist (SR3146.9.2.9)** UI element (page 140)
  (see **Workbench Object Operations - Smart Replace (SR3146.9.10.10)** operation (page 38))
- Delete
- **Status Handling of Custom Building Blocks (SR3146.10.1)** operation (page 87)
- Add <custom building block> to Setlist (via context menu for tab titles of the Graph Window of the **Work Area (SR3146.9.2.1)** UI element (page 136))
- Graph pagination
- **Usage List (SR3146.10.3)** operation (page 88)
- **Statistics (SR3146.9.2.18)** operation (page 161)
- Close (current or all)

The following operations are supported for system building blocks. They apply to elements on the phase level.

- Open existing (see **Open (SR3146.9.2.20+)** dialog (page 136))
- Save as new custom BB
- Close (current or all)
- **Compile Usage List of a System Phase (SR3146.10.6)** operation (page 90)
- **Statistics (SR3146.9.2.18)** operation (page 161)

- A building block (procedure, unit procedure, operation, phase) consists of a header and its underlying graphs, except for phases since they represent the lowest level.
  Available revisions of custom building blocks starting with the same identifier are listed for information.
- While renaming a custom BB: Available revisions of custom building blocks starting with the same identifier are listed for information.
- If a custom BB is copied (save as), the new building block is available for selection from the **Universe (SR3146.9.2.4+)** UI element (page 137) and the **Setlist (SR3146.9.2.9)** UI element (page 140) and is opened in the current tab. Available revisions of custom building blocks starting with the same identifier are listed for information.

Workbench Object Operations - New Custom Building Block Dialog (SR3146.9.10.7.2)

When a new custom building block is created, first, the building block author defines the building block identifier and revision.
The system shall allow to select an access privilege for the protection of the building block from unauthorized access for users with access privileges that allow to maintain protected building blocks. The list of available access privileges shall be restricted to the access privileges of the logged-in user.

If the new building block is an operation or unit procedure, the recipe author can decide with the **Usage as group** attribute if the building block can be used as a group:

- **Not allowed** (cannot be used as a group of building blocks)
- **Mandatory** (must be used as a group of building blocks)
- **Allowed** (can be used as a group of building blocks)

The **Usage as group** attribute can be changed as long as the building block is not in a read-only status.

Workbench Object Operations - View Locked Custom Building Block (SR3146.9.10.7.1)

If a custom building block is locked by another user, the system allows to view the building block without the possibility to change it.

The custom building block can be copied (save as). Operations that do not change a building block are also available (e.g. graph pagination, usage list, statistics).

Status Handling of Custom Building Blocks (SR3146.10.1)

Recipe and Workflow Designer provides a status change function for custom building blocks (**Draft**, **Verification**, **Approved**, **Archived**). This includes a check that the custom building blocks have no error or warning messages left in the **Messages (SR3146.9.2.16+)** window (page 144).

The system automatically locks all transitions without an identifier when a building block's status is changed to a read-only status.

Building blocks in the **Archived** status are available in the **Universe (SR3146.9.2.4+)** UI element (page 137) (of Recipe and Workflow Designer) and the Setlist **(SR3146.9.2.9)** UI element (page 140), but they cannot be added to a recipe, workflow, or custom building block.

- **Draft**: Initial status of building blocks.
- **Verification**: Building block is ready for review. If updates are required, a status change back to **Draft** has to be performed.
- **Approved**: Building blocks can no longer be modified. If an **Approved** building block is used within a master recipe, master workflow, or a custom building block, parameters or transitions that were locked are frozen and cannot be unlocked. Parameters or transitions that are not locked can still be modified. Underlying graphs are read-only. See "Locking of Parameters (SR3146.9.13+)" (page 31) and "Locking of Transitions (SR3146.9.14+)" (page 32).
- **Archived**: They are still available in the **Universe (SR3146.9.2.4+)** UI element (page 137) (of Recipe and Workflow Designer) and the **Setlist (SR3146.9.2.9)** UI element (page 140), but can only be added to a **change request SR3146.12+** object (page 97) (as old building block) from the Setlist. They are accessible via the **Open (SR3146.9.2.20+)** function (page 136).

Each status change must be confirmed by a single electronic signature.

The fixed state model supports the following status changes:

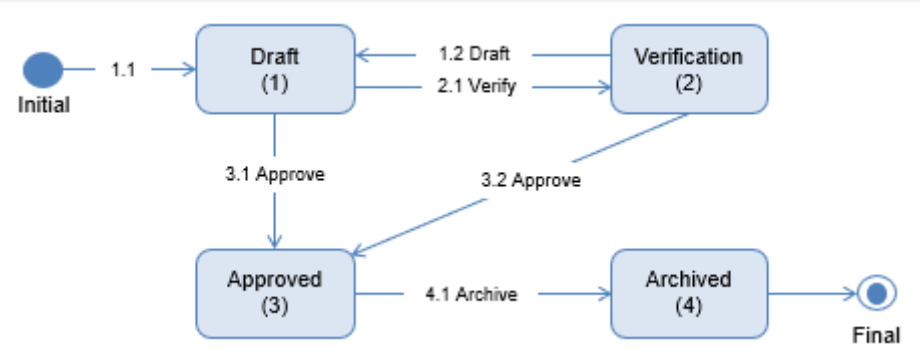


Figure 5: Building block - status graph

Transition (ID - From » To)	Signature (Access privilege)	Additional information
1.1 - Initial » Draft	---	Automatic transition.
1.2 - Verification » Draft	red_Status_Trans_Verification-Draft	---
2.1 - Draft » Verification	red_Status_Trans_Draft-Verification	---
3.1 - Draft » Approved	red_Status_Trans_Draft-Approved	---
3.2 - Verification » Approved	red_Status_Trans_Verification-Approved	---
4.1 - Approved » Archived	red_Status_Trans_Approved-Archived	---

Status History (SR3146.10.5)

Recipe and Workflow Designer provides a status history function for custom building blocks in order to track status transitions.
The history includes transitions that have taken place in a different environment (e.g. prior to export or import of a custom building block).

The history of status transitions executed before the introduction of the history with PharmaSuite 10.0 is generated automatically from the status data that is documented with the **Header Attributes (SR3146.10.4)** data (page 83). It is generated when the history is opened for the first time.

Revisioning (SR3146.10.2)

Revisioning of custom building blocks is supported as a mandatory free-text property.

Unique revisioning is requested when a building block is created (see "Workbench Object Operations (SR3146.9.10.7)" (page 84), "Workbench Object Operations - Element (SR3146.9.10.6)" (page 36)).

Usage List (SR3146.10.3)

The usage list of a custom building block provides all objects that have been designed based on the custom building block in question.
The following object types are supported by the usage list: custom building blocks, master recipes, and master workflows.
For objects that are protected by an access privilege: The system shall only provide objects whose access privilege matches the access privilege of the logged-in user.

Content (SR3146.10.3.1)

The usage list allows to track:

- Direct usage of a custom building block
  All objects that contain elements that were derived directly from the custom building block in question, without using an intermediate-level-approved custom building block.
  Example: A custom phase was built directly into a master recipe.
- Indirect usage of a custom building block
  All objects that contain elements that were not derived directly from the custom building block in question, but that were derived from an intermediate-level-approved custom building block.
  Example: A custom phase is used to build and approve a custom operation, which then was built directly into a master recipe.

Objects Information (SR3146.10.3.2)

Each object that is listed in the usage list provides the following information:

- Object type
- Identifier, version/revision, and status of the object
- Usage (direct, indirect)
- Used building block level
- As group
- [Used building block] Identifier, revision, and status
  - Direct usage: Identifier of the custom building block for which the usage list has been compiled.
  - Indirect usage: Identifier of the intermediate-level-approved custom building.
- Path information of the recipe, workflow, custom building block elements
  If the path belongs to a group, the last element identifier is that of the first element of the group.

Additionally, the usage list provides a list of changed objects whose objects information cannot be refreshed.

Filter (SR3146.10.3.3)

The usage list supports the following filter options:

- Usages [Any (default), All, Direct, Indirect]
- Status category [Any, All, Edit, Verification, Valid (default), Retired]

Usage List of a System Phase (SR3146.10.6)

The usage list of a system phase provides all objects that have been designed to use the system phase in question.
The following object types are supported by the usage list: custom building blocks, master recipes, and master workflows.
For objects that are protected by an access privilege: The system shall only provide objects whose access privilege matches the access privilege of the logged-in user.

Content (SR3146.10.6.1)

The usage list allows to track:

- Direct usage of a system phase
  All objects that contain the system phase in question, without using an intermediate-level-approved custom building block.
  Example: A system phase was built directly into an operation of a master recipe.
- Indirect usage of a system phase
  All objects that contain the system phase in question, but within an element that was derived from an intermediate-level-approved custom building block.
  Example: A system phase is used to build and approve a custom operation, which then was built directly into a master recipe.

Objects Information (SR3146.10.6.2)

Each object that is listed in the usage list provides the following information:

- Object type
- Identifier, version, and status of the object
- Usage (direct, indirect)
- As group (always **No**)
- Used building block level
- [Used building block] Identifier, version, and status
  - Direct usage: Identifier of the system phase for which the usage list has been compiled.
  - Indirect usage: Identifier of the intermediate-level-approved custom building.
- Path information of the recipe, workflow, custom building block elements, system phase

Additionally, the usage list provides a list of changed objects whose objects information cannot be refreshed.

Filter (SR3146.10.6.3)

The usage list supports the following filter options:

- Usages [Any (default), All, Direct, Indirect]
- Status category [Any, All, Edit, Verification, Valid (default), Retired]

Chapter 5

Comparing Master Recipes, Master Workflows, and Custom Building Blocks (SR3146.15+)

PharmaSuite supports the approval of master recipes, master workflows, and custom building blocks that are based on an approved version with a comparison mechanism.

The comparison baseline of a master recipe, master workflow, or custom building block defines the object to be compared with the current object.
The baseline is a header attribute of the **master recipe (SR3146.9.3.1)** object (page 10), the **master workflow (SR3146.9.12.1)** object (page 12), and the custom **building block (SR3146.10.4)** object (page 83).

The comparison result is displayed in the **Comparison (SR3146.9.2.19+)** window (page 146). It informs a reviewer of the recipe, workflow, or custom building block of the differences found by the comparison mechanism. Thus, the reviewer gets a summary of the differences.

Comparison Algorithm (SR3146.15.1+)

The comparison algorithm covers attributes, parameters, procedural elements within their SFC context, MFC-related settings, and the execution-relevant layout of a master recipe, master workflow, or custom building block.

Comparing Header Attributes (SR3146.15.1.1+)

Master recipe-specific header attributes (SR3146.15.1.1.1)

Each attribute is compared with its baseline, except for:

- Identifier
- Version
- Comparison baseline
- Status
- Valid from and Valid until

Master workflow-specific header attributes (SR3146.15.1.1.2)

Each attribute is compared with its baseline, except for:

- Identifier
- Version
- Comparison baseline
- Status
- Valid from and Valid until

Custom building block-specific header attributes (SR3146.15.1.1.3)

Each attribute is compared with its baseline, except for:

- Identifier
- Revision
- Comparison baseline
- Usage as <phase or operation> group
- Status
- Approved on and Approved by
- Archived on and Archived by

Comparing Element Attributes (SR3146.15.1.2)

Each attribute is compared with its baseline.

Comparing Source Attributes (SR3146.15.1.3)

Only the **Custom** and **Group** source attributes and the **System** identifier attribute is compared with its baseline.

The following **System** attributes are not compared:

- Short description
- Usable in
- Trigger
- Server-run
- Pause-aware

Comparing Parameter Attributes (SR3146.15.1.4+)

Material parameter-specific attributes (SR3146.15.1.4.1)

Each attribute is compared with its baseline, except for:

- Material short description

Material Flow Control-specific attributes (SR3146.15.1.4.2)

Each attribute is compared with its baseline.

Equipment requirement parameter-specific attributes (SR3146.15.1.4.3)

Each attribute is compared with its baseline, except for:

- Short description
- Description
- EQM level

Work center assignment parameter-specific attributes (SR3146.15.1.4.4)

Each attribute is compared with its baseline, except for:

- Description
- Storage area

Privilege parameter-specific attributes (SR3146.15.1.4.5)

Each attribute is compared with its baseline, except for:

- Default reason (1st and 2nd)

Capability parameter-specific attributes (SR3146.15.1.4.6)

Each attribute is compared with its baseline.

Process parameter-specific attributes (SR3146.15.1.4.7)

Each attribute is compared with its baseline, except for:

- Content (concatenation of other attributes that are being compared)

Comparing Procedural Elements (SR3146.15.1.5+)

Transition-specific attributes (SR3146.15.1.5.2)

Each attribute is compared with its baseline.

Relation-specific characteristics (SR3146.15.1.5.1)

The following attributes are compared with its baseline:

- Existence of elements
- Successor relations of elements
- Branching type (sequence, selection, simultaneous)

Location-specific characteristics (SR3146.15.1.5.3)

The following attribute is compared with its baseline:

- Position (relative layout) of elements in simultaneous branches

The position of elements in simultaneous branches can affect the preview rendering during execution and the sequence of elements within the batch report.

Defining a Comparison Baseline (SR3146.15.2)

The system allows to define a comparison baseline for master recipes and master workflows in the **Edit** status (page 49) and custom building blocks in the **Draft** status (page 87).

The following rules apply to comparison baseline objects:

- For master recipes or master workflows, they must be in the **Scheduled**, **Valid**, or **Archived** status (page 49).
- For custom building blocks, they must be in the **Approved** or **Archived** status (page 87).

Performing a Comparison (SR3146.15.3)

To perform a comparison, a comparison baseline must be assigned. As long as no comparison baseline is assigned, the system displays an appropriate message.

When a comparison has been performed, the system displays the result in the list panel of the **Comparison (SR3146.9.2.19+)** window (page 146).

- If no differences have been detected, the system displays an appropriate message.
- If differences have been detected, the system displays the details.
- If the current object has been changed in the meantime at the current client, the system displays an appropriate message.

Chapter 6

Management of Change Requests (SR3146.12+)

A change request consists of its **header attributes (SR3146.12.1)** data (page 97), the **scope definition (SR3146.12.3+)** feature (page 106), and the **action list (SR3146.12.4)** feature (page 109).

The scope of a change request is defined by the old (to be replaced) and new (replacing) building blocks and by the master recipes/master workflows and custom building blocks selected from the usage list that shall be subject to changing.
The action list shows all master recipes/master workflows and custom building blocks which are to be changed.

During the **execution of a change request (SR3146.12.5)** operation (page 110), the replacement of building blocks follows the same rules as for the **Smart Replace (SR3146.9.10.10)** operation (page 38).

The life cycle of a change request is controlled by the **status management of change requests (SR3146.12.2+)** feature (page 99).

The change request can be scheduled to be executed in background by the **Scheduled Execution of Change Requests (SR3146.12.7)** operation (page 113).

Header Attributes (SR3146.12.1)

The change request object supports the following header attributes:

- Basic data:
  - Identifier
  - Description
  - Comparison baseline handling (Keep, Update, Delete)
  - Planned execution time
- Recipe-/workflow-related data:
  - Target status (Edit, Verification, All valid statuses)
  - Effective date
- Building block-related data:
  - Target status (Draft, Verification, Approved)
- Status data:
  - Status

- Additional data:
  - Structure level
  - System building block (true/false)

The attributes are maintained in the **header component-specific (SR3146.9.2.15)** property window (page 144).

The **Target status** attributes define if the related new recipe or workflow versions are automatically moved to the **Valid** status (page 49) and if the custom building block revisions are automatically moved to the **Approved** status (page 87), respectively.

The **Comparison baseline handling** attribute allows to

- keep the comparison baseline of the old component to also be the baseline of the new component;
- update (default setting) the comparison baseline of the new component according to the most recent version/revision from which it was created;
- delete the comparison baseline, so that for the new component no comparison baseline is defined even if a comparison baseline was defined for the old component.

Workbench Object Operations (SR3146.12.6)

The following operations are supported for change requests:

- Create new change request
- Open existing (see **Open (SR3146.9.2.20+)** dialog (page 136))
- View locked change request (page 99)
- Save (current or all)
- Copy (Save as...)
- Rename
- Delete
- Close (current or all)

Open Dialog - Hide Objects (SR3146.12.6.3)

By default, change requests that are **Finished** or in a **Retired** status (e.g. **Archived**, **Obsolete**) are excluded from the search result.

The inclusion of change requests that are **Finished** or in a **Retired** status can be switched on.

Open Dialog for Protected Objects (SR3146.12.6.2)

A change request can hold objects that are protected by an access privilege, like a new or old custom building block or any object that is in the scope of the change request according to the action list (master recipes, master workflows, and custom building blocks). In case a protected object is involved: The system shall only allow to open change requests whose access privileges of any of the protected objects match the access privilege of the logged-in user.

Workbench Object Operations - View Locked Change Request (SR3146.12.6.1)

If a change request is locked by another user, the system allows to view the change request without the possibility to change it.

The change request can be copied (save as). Operations that do not change a change request are also available (e.g. status history).

Status Management of Change Requests (SR3146.12.2+)

Master Recipe-specific Change Requests (SR3146.12.2.2)

The **version control (SR3146.4.5)** function (page 161) supports the following default transitions for master recipe-specific change requests:

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
1.1 - Initial » Draft	---	Automatic transition.
1.2 - In review » Draft	Status_Trans_ChangeRequest_InReview-Draft	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
2.1 - Draft » In review	Status_Trans_ChangeRequest_Draft-InReview	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor) Completeness of old/new building block pairs is checked. If the action list contains an object of the **Master Recipe** type, the target status for **Recipe-related Data** needs to be set. If the action list contains an object of any building block type, the target status for **Building Block-related Data** needs to be set.

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
3.1 - In review » Approved for auto-run	Status_Trans_ChangeRequest_InReview-ApprovedAutoRunnable	Default user group: Qualified Person Planned execution time needs to be set and be equal to or earlier than the effective date.
4.1 - In review » Approved	Status_Trans_ChangeRequest_InReview-Approved	Default user group: Qualified Person
4.2 - Approved for auto-run » Approved	Status_Trans_ChangeRequest_ ApprovedAutoRunnable-Approved	Default user group: Qualified Person
5.1 - Approved » In process	Status_Trans_ChangeRequest_Approved-InProcess	Automatic transition when the change request is being executed. Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
5.2 - Approved for auto-run » In process	---	Automatic transition when the change request is being executed by the background job.
6.1 - In process » Finished	---	Automatic transition when the change request execution is completed.
6.2 - Aborted » Finished	Status_Trans_ChangeRequest_Canceled-Finished	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
6.3 - In process » Finished	Status_Trans_ChangeRequest_InProcess-Finished	Manual transition, only available if the execution has been interrupted (e.g. resume during execution). Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
7.1 - Finished » Archived	---	---
7.2 - Obsolete » Archived	---	---
8.1 - Draft » Obsolete	---	---
8.2 - In review » Obsolete	Status_Trans_ChangeRequest_InReview-Obsolete	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
8.3 - Approved » Obsolete	Status_Trans_ChangeRequest_Approved-Obsolete	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
8.4 - Aborted » Obsolete	Status_Trans_ChangeRequest_Canceled-Obsolete	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
8.5 - In process » Obsolete	Status_Trans_ChangeRequest_InProcess-Obsolete	Manual transition, only available if the execution has been interrupted (e.g. resume during execution). Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
8.6 - Approved for auto-run » Obsolete	Status_Trans_ChangeRequest_ApprovedAutoRunnable-Obsolete	Default user group: Qualified Person
8.7 - Auto-run failed » Obsolete	Status_Trans_ChangeRequest_AutoRunnableFailed-Obsolete	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
9.1 - In process » Aborted	Status_Trans_ChangeRequest_InProcess-Canceled	Automatic transition when the change request is canceled with the ESC key.
10.1 - In process » Auto-run failed	---	Automatic transition when the change request executed by the background job failed.

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

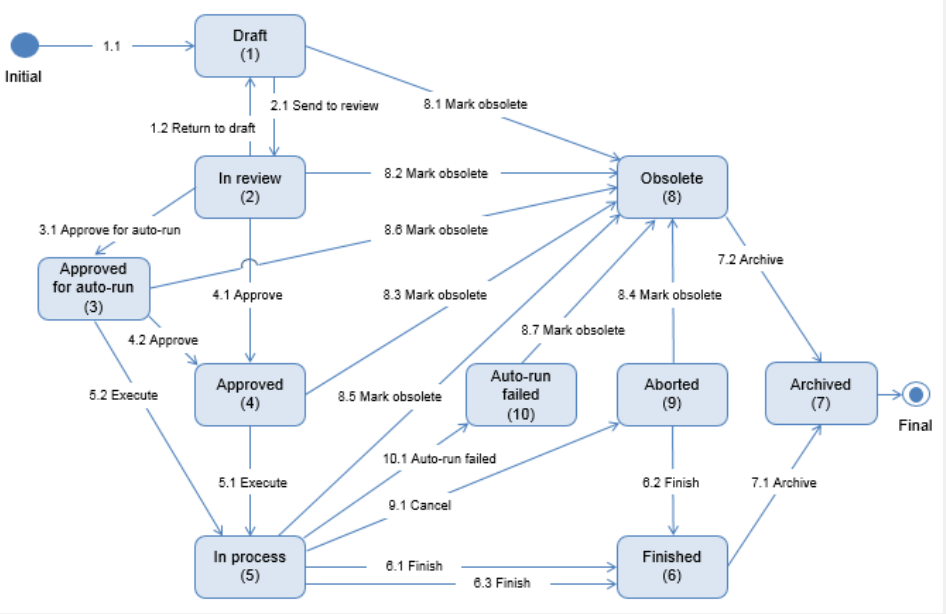


Figure 6: Master recipe-specific change request - status graph

Master Workflow-specific Change Requests (SR3146.12.2.3)

The **version control (SR3146.4.5)** function (page 161) supports the following default transitions for master workflow-specific change requests:

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
1.1 - Initial » Draft	---	Automatic transition.
1.2 - In review » Draft	Status_Trans_ChangeRequest_WF_InReview-Draft	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
2.1 - Draft » In review	Status_Trans_ChangeRequest_WF_Draft-InReview	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor) Effective date in the past is not allowed. Completeness of old/new building block pairs is checked. If the action list contains an object of the **Master Workflow** type, the target status for **Workflow-related Data** needs to be set. If the action list contains an object of any building block type, the target status for **Building Block-related Data** needs to be set.
3.1 - In review » Approved for auto-run	Status_Trans_ChangeRequest_WF_InReview-ApprovedAutoRunnable	Default user group: Qualified Person Planned execution time needs to be set and be equal to or earlier than the effective date.
4.1 - In review » Approved	Status_Trans_ChangeRequest_WF_InReview-Approved	Default user groups: Workflow Author, Qualified Person Effective date in the past is not allowed.
4.2 - Approved for auto-run » Approved	Status_Trans_ChangeRequest_WF_ApprovedAutoRunnable-Approved	Default user group: Qualified Person
5.1 - Approved » In process	Status_Trans_ChangeRequest_WF_Approved-InProcess	Automatic transition when the change request is being executed. Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
5.2 - Approved for auto-run » In process	---	Automatic transition when the change request is being executed by the background job.
6.1 - In process » Finished	---	Automatic transition when the change request execution is completed.
6.2 - Aborted » Finished	Status_Trans_ChangeRequest_WF_Canceled-Finished	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
6.3 - In process » Finished	Status_Trans_ChangeRequest_WF_InProcess-Finished	Manual transition, only available if the execution has been interrupted (e.g. resume during execution). Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
7.1 - Finished » Archived	---	---
7.2 - Obsolete » Archived	---	---
8.1 - Draft » Obsolete	---	---
8.2 - In review » Obsolete	Status_Trans_ChangeRequest_WF_InReview-Obsolete	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
8.3 - Approved » Obsolete	Status_Trans_ChangeRequest_WF_Approved-Obsolete	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
8.4 - Aborted » Obsolete	Status_Trans_ChangeRequest_WF_Canceled-Obsolete	Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)

Transition (ID - From » To)	Signature (Access privilege, according to FSM)	Additional information
8.5 - In process » Obsolete	Status_Trans_ChangeRequest_WF_InProcess-Obsolete	Manual transition, only available if the execution has been interrupted (e.g. resume during execution). Default user groups: Workflow Author, Qualified Person, Supervisor (Shop Floor)
8.6 - Approved for auto-run » Obsolete	Status_Trans_ChangeRequest_WF_ApprovedAutoRunnable-Obsolete	Default user groups: Workflow Author, Qualified Person
8.7 - Auto-run failed » Obsolete	Status_Trans_ChangeRequest_WF_AutoRunnableFailed-Obsolete	Default user groups: Recipe Author, Qualified Person, Supervisor (Shop Floor)
9.1 - In process » Aborted	Status_Trans_ChangeRequest_WF_InProcess-Canceled	Automatic transition when the change request is canceled with the ESC key.
10.1 - In process » Auto-run failed	---	Automatic transition when the change request executed by the background job failed.

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

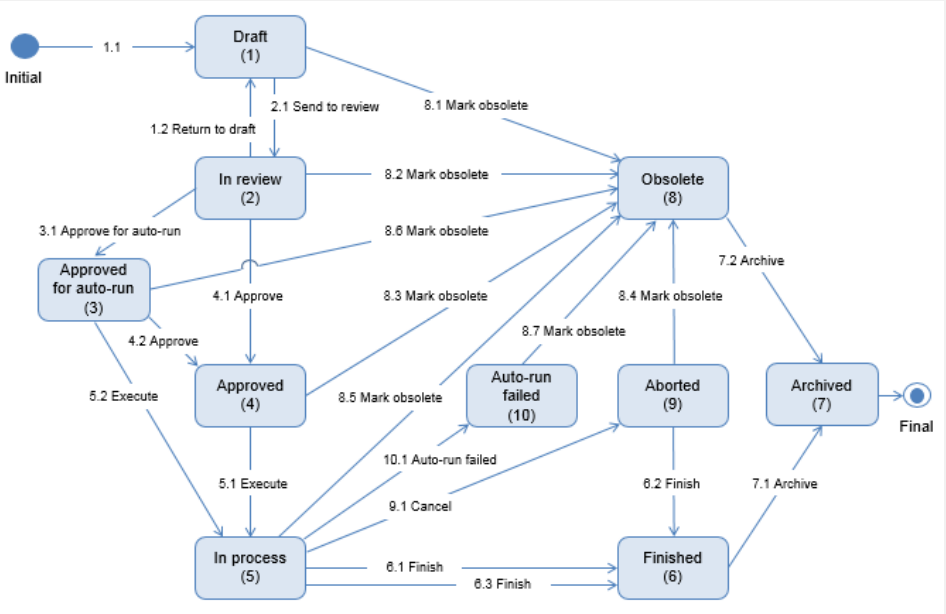


Figure 7: Master workflow-specific change request - status graph

Scope Validity Check (SR3146.12.2.1)

During the status transitions of a change request to **In review**, **Approved**, and **In process**, the system checks if the defined scope of the change request is still valid. This applies to the status of the new building blocks and to the status/version of any affected object (master recipe, master workflow, or custom building block).

Change Status Dialog (SR3146.12.2.4)

Errors occurring during the status transition are displayed and can be copied to the clipboard.

Scope Definition (SR3146.12.3+)

The scope definition of a change request is composed of the **definition of old/new building blocks (SR3146.12.3.1)** task (page 107) and the **selection from the usage list (SR3146.12.3.2)** task (page 108).

Define Old/New Building Blocks (SR3146.12.3.1)

The system allows to define pairs of old and new building blocks for a change request. In case the new building block contains unresolved external references at locked parameters, the system displays an information message that the external references will be resolved during execution of the change request. During the execution of the change request, the old building blocks are replaced by the new building blocks.

The following rules apply:

- All building block levels of the procedural structure (page 14) are supported (procedure, unit procedure, operation, phase).
- The first selected old or new building block (group) defines the structure level of the change request. It is stored with the change request header data and visible in the column headers of the building block table. All further selected building blocks (building block groups) must have the same structure level.
- Operation groups belong to the structure level of unit procedures and phase groups to the structure level of operations.
- On the structure levels of unit procedures and operations, the old and new building blocks (building block groups) additionally display their **Usage as group** setting.
- An old building block (group) must be either in the **Approved** or the **Archived** statuses (page 87).
- A new building block (group) must be in the **Approved** status (page 87).
- A building block can only be selected once as old building block.

  TIP
  An exception to this rule is a building block with **Allowed** as **Usage as group** setting, which may be used twice.

- An old building block (group) cannot also be a new building block (group) of the same or any other old/new pair of the change request.
- An old building block group with **Mandatory** as **Usage as group** setting requires a new building block with either **Allowed** or **Mandatory** as its **Usage as group** setting.
- An old building block group with **Not allowed** as **Usage as group** setting requires a new building block with either **Allowed** or **Not allowed** as its **Usage as group** setting.
- The system allows to remove an old/new building block pair from the list.
- After removing all old/new building block pairs from the list, the structure level of the change request is reset.

- The access privilege for the protection of a custom building block from unauthorized access has no impact on the replacement. It is always performed.

To allow greater flexibility when defining the old and new building block pairs, the following rules are checked later during the next change request status transition and can also depend on the occurrences selected in the usage list:

- If a building block with **Allowed** as **Usage as group** setting is selected as old building block and in the usage list, there is an element selected that uses the old building block as group, the new building block must have either A**llowed** or **Mandatory** as its **Usage as group** setting.
- If a building block with **Allowed** as **Usage as group** setting is selected as old building block and in the usage list, there is an element selected that uses the old building block as stand-alone unit procedure or operation, the new building block cannot have **Mandatory** as **Usage as group** setting.
- If an old building block contains a group, this group cannot be selected as old building block in another replacement pair.
- If a building block with **Allowed** as **Usage as group** setting is selected twice as old building block, one replacement pair must hold a new building block with **Not allowed** as **Usage as group** setting, while the other replacement pair must hold a new building block with **Mandatory** as **Usage as group** setting.

Select from Usage List (SR3146.12.3.2)

Based on the definition of the old building block, the system allows to compile a usage list **(SR3146.10.3)** for custom building blocks (page 88) and also a usage list **(SR3146.10.6)** for system phases (page 90) to select a subset of artifacts to be changed within the scope of the change request.

The following rules apply:

- The **Compile usage list** function is only available for change requests in the **Draft** status (page 99).
- The usage list only includes master recipes and master workflows in the **Valid** or **Scheduled** statuses (page 49) and building blocks in the **Approved** or **Archived** statuses (page 87).
  Only those usages are displayed for which the old building block was in the **Approved** status (page 87) when it was copied (i.e. **Draft** usages are filtered out).
- If the status of a master recipe, master workflow, or building block has changed since the last compilation of the usage list and the change request is still in the **Draft** status, the system notifies the user of the status change and updates the usage list. Example: Master recipes whose status has been changed to **Archived** are removed from the list.

- If an object (e.g. master recipe) selected for a change includes indirect usages of the old building block, in the usage list, the system selects and marks the affected custom building blocks to be changed within the scope of the change request.
- If an object (e.g. master recipe) includes indirect usages of the old building block and the object is unselected, in the usage list, the system unselects the affected custom building blocks that were previously selected due to their indirect usage.
- The system indicates visually if an object was selected manually, due to an indirect usage, or both.
- For objects that are protected by an access privilege: The system shall only provide objects whose access privilege matches the access privilege of the logged-in user.

The following filter attributes can be set when compiling the usage list:

- Material (product): Only master recipes and their directly or indirectly used objects are displayed if the master recipe's material (product) identifier matches the filter criteria.
  The S88DefaultWorkflowMaterial can be used for master workflows to display only master workflows and directly and indirectly used objects of the master workflows.
- Equipment class: Only objects are displayed that, in any of their elements, use the selected equipment class(es) as equipment requirement or use the selected equipment class(es) within an expression that contains the **Equipment is member of class** function. These objects, objects that directly or indirectly use these objects, or objects that are used by these objects are displayed.
- Work center: Only objects with unit procedures, unit procedures that have a work center assignment that matches the filter criteria, or objects that are used by these objects are displayed.

Action List (SR3146.12.4)

The action list represents all master recipes, master workflows, and custom building blocks that are in the scope of the change request, i.e., that are selected in the usage list and hence require a new version/revision due to the defined building block replacement.

The action list provides the following information for each artifact:

- Object type
- Identifier
- Old object [version/revision and status before and after executing the change request]
- New object [version/revision and status]

- Progress
- Result
- Error

An artifact of the action list represents one or more of the defined pairs of old/new building blocks.

It is possible to define a **planned execution time** for the change request in the **Draft** status if the change request shall be processed in the background. For details, please refer to the **Scheduled Execution of Change Requests (SR3146.12.7)** requirement (page 113).

Execute Change Request (SR3146.12.5)

During the automatic execution of a change request, new versions/revisions of the affected master recipes/master workflows/custom building blocks are created and the defined building blocks are replaced.

Only one new version/revision is created per action list object, regardless of how many replacements of different building blocks are performed according to the old/new building block pairs definitions. If one of the replacements fails with an error, no new version/revision of the affected object is created.

Related to the change request, the following rules apply:

- Only change requests in the **Approved** status (page 99) can be executed.
- The system sets the status of a change request automatically to **In process** when the change request is being executed.
- The system sets the status of a change request automatically to **Finished** when the change request execution is completed.

Related to the artifacts displayed in the **Action list (SR3146.12.4)** tab (page 109), the following rules apply:

- The action list is processed bottom-up (first artifacts on system phase level, last the master recipes/workflows).
- The replacement of building blocks follows the rules of the **Smart Replace (SR3146.9.10.10)** operation (page 38).
- The system tries to automatically resolve external references in expressions. Each resolved reference is indicated by an information message that is stored with the master recipe or master workflow.

- The system automatically transitions new revisions of building blocks to the target status that is defined with the building block-related data in the **header attributes (SR3146.12.1)** data (page 97) of the change request. In case the old object status is **Archived**, the new object status is **Archived** again, regardless of the target status definition.
- Master recipes and master workflows in the **Valid** or **Scheduled** status (page 49) are all considered to represent valid recipes and workflows. New versions will be transitioned according to their originally defined period of validity.
- The system automatically transitions new versions of master recipes and master workflows to the target status that is defined with the recipe-/workflow-related data in the **header attributes (SR3146.12.1)** data (page 97) of the change request.
- The following descriptions reflect that the **Target status** attribute has been set to **All valid statuses**.
  (If it has been set to **Edit** or **Verification**, new versions of master recipes and master workflows in the **Edit** or **Verification** statuses are created. Old versions of affected master recipes or master workflows are not changed, i.e. their statuses remain unchanged.)
  - If the effective date of the change request is not defined (or today), an affected master recipe or master workflow in the **Valid** status is archived and a new version of the master recipe or master workflow in the **Valid** status is created.
  - If the effective date of the change request is in the future and is due while an affected master recipe or master workflow is in the **Valid** status, the master recipe's or master workflow's period of validity is shortened and a new version of the master recipe or master workflow in the **Scheduled** status is created. For this new version in the **Scheduled** status, the period of validity starts on the effective date of the change request and lasts until the end of the original period of validity.
  - If the effective date of the change request is due before the period of validity of an affected master recipe or master workflow in the **Scheduled** status starts, a new version of the master recipe or master workflow is created in the **Scheduled** status based on the original period of validity of the previous version.
  - If the effective date of the change request is in the future and is due while an affected master recipe or master workflow is in the **Scheduled** status, the master recipe's or master workflow's future period of validity is shortened and another version of the master recipe or master workflow in the **Scheduled** status is created. For this new version, the period of validity starts on the effective date of the change request and lasts until the end of the original period of validity.

- For all status transitions related to master recipes and master workflows that require an electronic signature, a silent signature is recorded whose comment is filled with the data of the signing operator(s) that approved the change request.
- In case an error occurs during execution of a change request, the result for the current artifact is recorded in the **Action list (SR3146.12.4)** tab (page 109) and the system proceeds with the next artifact to be changed.
  Examples for error conditions: Status has changed since the compilation of the usage list, procedural structure is broken, unresolved external references cannot be resolved, etc.

- As only master recipes and master workflows in the **Valid** or **Scheduled** status (page 49) can be part of the action list, master recipes and master workflows can never be transitioned by a change request to a status that is higher than the original master recipe's or master workflow's status prior to the execution of the change request.
- For all status transitions of master recipes and master workflows that occur in the context of a change request execution, the master recipe- and master workflows-specific configurations related to signature privileges and e-mail notifications are ignored by the system.

Execution Status (SR3146.12.5.2)

The change request supports the following interim execution statuses:

- Not started
- Processing
- Canceling (triggered by the ESC key)

The change request supports the following final execution statuses:

- Canceled (subsequent to the **Canceling** interim execution status)
- Canceled (unexpected, after a resume of a running change request)
- Completed

Cancel Change Request (SR3146.12.5.1)

During the automatic execution of a change request, the execution of the change request can be canceled.

The following rules apply:

- The system completes the replacement-related steps of the artifacts that are displayed in the **Action list (SR3146.12.4)** tab (page 109) and are currently in process.
- All artifacts displayed in the **Action list (SR3146.12.4)** tab (page 109) that have not been started yet are marked as **Canceled**.
- The system sets the status of the change request automatically to **Aborted** when the execution of the change request has been canceled.

Scheduled Execution of Change Requests (SR3146.12.7)

Change requests in the **Approved for auto-run** status are monitored by the system and executed if its planned execution time has passed. The execution is performed as described with the **Execute Change Request (SR3146.12.5)** operation (page 110).

If the action list contains an error for any artifact, the change request is set to the **Auto-run failed** status instead of the **Finished** status.

Chapter 7

Master Recipe Report - Batch (SR3146.11+)

The master recipe report - batch represents the S88-compliant structure of a master recipe and its components in a printable way.

Print MR Report (SR3146.11.1)

The system allows the user to print a master recipe report that represents the S88-compliant structure of a master recipe. If desired, the comparison result can be excluded from the report.

MR Report - SFC Pagination (SR3146.11.1.1)

Displays the current graph structure (SFC flow) in a preview window for printing including pagination preview.

MR Report Header (SR3146.11.2)

The header of the MR report includes the following data:

- Placeholder for a customer logo
- Material ID and Short description
- MR identifier
- MR version

MR Report Footer (SR3146.11.3)

The footer of the MR report includes the following data:

- Rockwell Automation logo
- Printed by: <user name and login>
- Printed from: <station/work center>

- Database: <database name>
- Printed on: <local time and server time>
- Page x of y

MR Report Cover Page (SR3146.11.5)

The cover page of the MR report includes the following data:

- Report title
- Identifier
- Description
- Comparison baseline
- Material identifier and Short description
- Planned quantity
- Version and Status
- Valid from and Valid until
- Method and Method description

MR Report Sections (SR3146.11.4+)

The MR report includes the following sections:

MR report - Table of contents (SR3146.11.4.15)

The table of contents of the MR report.

MR report - Basic data (SR3146.11.4.1)

Section for the basic data of the master recipe, which includes:

- Identifier
- Description
- Comparison baseline
- Material identifier and Short description
- Planned quantity
- Minimum quantity and Maximum quantity
- Version and Status
- Valid from and Valid until

- Registration number
- Method and Method description
- Usage type
- Review mode
- Access privilege
- Packaging levels

MR report - Comments (SR3146.11.4.2)

Section for the comments related to the master recipe.

MR report - Approval record (SR3146.11.4.3)

Section for the approval record of the current version of the master recipe, which includes:

- For the creation of the recipe:
  Action, role, logged-in user, reason for creation
- For the scheduling of the recipe:
  Action, role, signatures with timestamp, signature comment
- For the automatic approval of the recipe due to schedule:
  Action, user with timestamp
- For the approval of the recipe:
  Action, role, signatures with timestamp, signature comment

MR report - Bill of materials (SR3146.11.4.4)

Formula section contains the bill of materials of the master recipe, which includes the following header attributes and columns:

- Header attributes:
  - Material [identifier and short description]
  - ERP BOM alternative
  - ERP BOM base quantity
  - ERP BOM released
- Columns:
  - Position
  - Material [identifier and short description]

- Material type
- Planned quantity
- Fixed quantity

MR report - Work center assignments (SR3146.11.4.5)

Formula section contains the list of work centers of the master recipe, which includes the following columns:

- Type
- Identifier
- Description
- Used in recipe element

MR report - Material flow control (SR3146.11.4.6)

Formula section contains the material flow control data of the master recipe, which includes the following columns:

- MFC item
- Position
- Material [identifier and short description]
- Planned quantity (input)
- Planned quantity (output)
- Output unit procedure
- Input unit procedure
- Target MFC item

MR report - Critical process parameters (SR3146.11.4.7)

Formula section contains the critical process parameters of the master recipe, which includes the following columns:

- Identifier
- Content
- Path

MR report - Critical quality attributes (SR3146.11.4.8)

Formula section contains the critical quality attributes of the master recipe, which includes the following columns:

- Identifier
- Content
- Path

MR report - Key process parameters (SR3146.11.4.9)

Formula section contains the key process parameters of the master recipe, which includes the following columns:

- Identifier
- Content
- Path

MR report - Key quality attributes (SR3146.11.4.10)

Formula section contains the key quality attributes of the master recipe, which includes the following columns:

- Identifier
- Content
- Path

MR report - Procedure (SR3146.11.4.11)

The procedure-related data of the MR report includes the following details:

- Basic information
  - Element data
  - Custom source data
- List of steps
- List of transitions
- Graphical representation of the SFC flow

MR report - Unit procedure (SR3146.11.4.12)

For each unit procedure, the MR report includes the following details:

- Basic information
  - Element data
  - Custom source data
- Work center information
- Capability information (e.g. Detachable capability, Pause-enabled capability)
- Privilege information
- List of steps
- List of transitions
- List of used groups (only operation groups used in a read-only status)
- Graphical representation of the SFC flow

MR report - Operation (SR3146.11.4.13)

For each operation, the MR report includes the following details:

- Basic information
  - Element data
  - Custom source data
  - Group source data
- Station information
- Capability information (e.g. Detachable capability, Event-triggered capability)
- Privilege information
- List of steps
- List of transitions
- List of used groups (only phase groups used in a read-only status)
- Graphical representation of the SFC flow

MR report - Phase (SR3146.11.4.14)

For each phase, the MR report includes the following details:

- Basic information
  - Element data
  - Custom source data
  - Group source data
  - System source data
- Input material information
- Dispensing-specific material information (only for Dispense operations)
- Output material information
- Equipment requirement information
- Privilege information
- Process parameter information

MR report - Comparison with baseline (SR3146.11.4.17)

For each comparison-specific type, the MR report includes the following details:

- Mode
- Path
- Type
- Identifier
- Predecessor
- Successor
- Attribute
- New value
- Old value

If no differences have been detected between the master recipe and its comparison baseline, no comparison baseline is defined, or the comparison baseline is not available, the report includes an appropriate message instead of the table.

Chapter 8

Master Workflow Report (SR3146.13+)

The master workflow report represents the S88-compliant structure of a master workflow and its components in a printable way.

Print MWF Report (SR3146.13.1)

The system allows the user to print a master workflow report that represents the S88-compliant structure of a master workflow. If desired, the comparison result can be excluded from the report.

MWF Report - SFC Pagination (SR3146.13.1.1)

Displays the current graph structure (SFC flow) in a preview window for printing including pagination preview.

MWF Report Header (SR3146.13.2)

The header of the MWF report includes the following data:

- Placeholder for a customer logo
- Processing name
- Master workflow identifier and Short description
- Master workflow version

MWF Report Footer (SR3146.13.3)

The footer of the MWF report includes the following data:

- Rockwell Automation logo
- Printed by: <user name and login>
- Printed from: <station/work center>

- Database: <database name>
- Printed on: <local time and server time>
- Page x of y

MWF Report Cover Page (SR3146.13.5)

The cover page of the MWF report includes the following data:

- Report title
- Identifier
- Processing name
- Short description
- Version and Status
- Valid from and Valid until

MWF Report Sections (SR3146.13.4+)

The MWF report includes the following sections:

MWF report - Table of contents (SR3146.13.4.13)

The table of contents of the MWF report.

MWF report - Basic data (SR3146.13.4.1)

Section for the basic data of the master workflow, which includes:

- Identifier
- Processing name
- Short description
- Description
- Comparison baseline
- Version and Status
- Valid from and Valid until
- Review mode
- Access privilege
- [Workflow] Type
- One-click startable

- Execution prefix
- Production-relevant
- Appendable during processing

MWF report - Comments (SR3146.13.4.2)

Section for the comments related to the master workflow.

MWF report - Approval record (SR3146.13.4.3)

Section for the approval record of the current version of the master workflow, which includes:

- For the creation of the workflow:
  Action, role, logged-in user, reason for creation
- For the scheduling of the workflow:
  Action, role, signatures with timestamp, signature comment
- For the automatic approval of the workflow due to schedule:
  Action, user with timestamp
- For the approval of the workflow:
  Action, role, signatures with timestamp, signature comment

MWF report - List of materials (SR3146.13.4.15)

Formula section contains the list of materials of the master workflow, which includes the following columns:

- Position
- Material [identifier and short description]
- Material type
- Planned quantity
- Fixed quantity

MWF report - Work center assignments (SR3146.13.4.4)

Formula section contains the list of work centers of the master workflow, which includes the following columns:

- Type
- Identifier
- Description
- Used in workflow element

MWF report - Critical process parameters (SR3146.13.4.5)

Formula section contains the critical process parameters of the master workflow, which includes the following columns:

- Identifier
- Content
- Path

MWF report - Critical quality attributes (SR3146.13.4.6)

Formula section contains the critical quality attributes of the master workflow, which includes the following columns:

- Identifier
- Content
- Path

MWF report - Key process parameters (SR3146.13.4.7)

Formula section contains the key process parameters of the master workflow, which includes the following columns:

- Identifier
- Content
- Path

MWF report - Key quality attributes (SR3146.13.4.8)

Formula section contains the key quality attributes of the master workflow, which includes the following columns:

- Identifier
- Content
- Path

MWF report - Procedure (SR3146.13.4.9)

The procedure-related data of the MWF report includes the following details:

- Basic information
  - Element data
  - Custom source data
- List of steps
- Graphical representation of the SFC flow

MWF report - Unit procedure (SR3146.13.4.10)

For each unit procedure, the MWF report includes the following details:

- Basic information
  - Element data
  - Custom source data
- Work center information
- Capability information (e.g. Detachable capability, Pause-enabled capability)
- Privilege information
- List of steps
- List of transitions
- List of used groups (only operation groups used in a read-only status)
- Graphical representation of the SFC flow

MWF report - Operation (SR3146.13.4.11)

For each operation, the MWF report includes the following details:

- Basic information
  - Element data
  - Custom source data
  - Group source data
- Station information
- Capability information (e.g. Detachable capability)
- Privilege information
- List of steps

- List of transitions
- List of used groups (only phase groups used in a read-only status)
- Graphical representation of the SFC flow

MWF report - Phase (SR3146.13.4.12)

For each phase, the MWF report includes the following details:

- Basic information
  - Element data
  - Custom source data
  - Group source data
  - System source data
- Input material information
- Output material information
- Equipment requirement information
- Privilege information
- Process parameter information

MWF report - Comparison with baseline (SR3146.13.4.17)

For each comparison-specific type, the MWF report includes the following details:

- Mode
- Path
- Type
- Identifier
- Predecessor
- Successor
- Attribute
- New value
- Old value

If no differences have been detected between the master workflow and its comparison baseline, no comparison baseline is defined, or the comparison baseline is not available, the report includes an appropriate message instead of the table.

Chapter 9

Designer Workbench (SR3146.9+)

➢ As requirements have been restructured compared to the original DRS document, this section also contains requirements of other requirement groups than SR3146.9+.

For recipe or workflow authors, the graphical Recipe and Workflow Designer workbench provides the means to create and maintain master recipes, master workflows, and custom building blocks (cBB).

This section provides the requirements related to Designer workbench itself:

Workbench Usability (SR3146.9.1+)

The graphical Designer workbench supports usability aspects related to its graphical user interface and general handling. Both are targeted to minimize time to result significantly.

Custom Building Block Image

The custom building block image provides access to the following data:

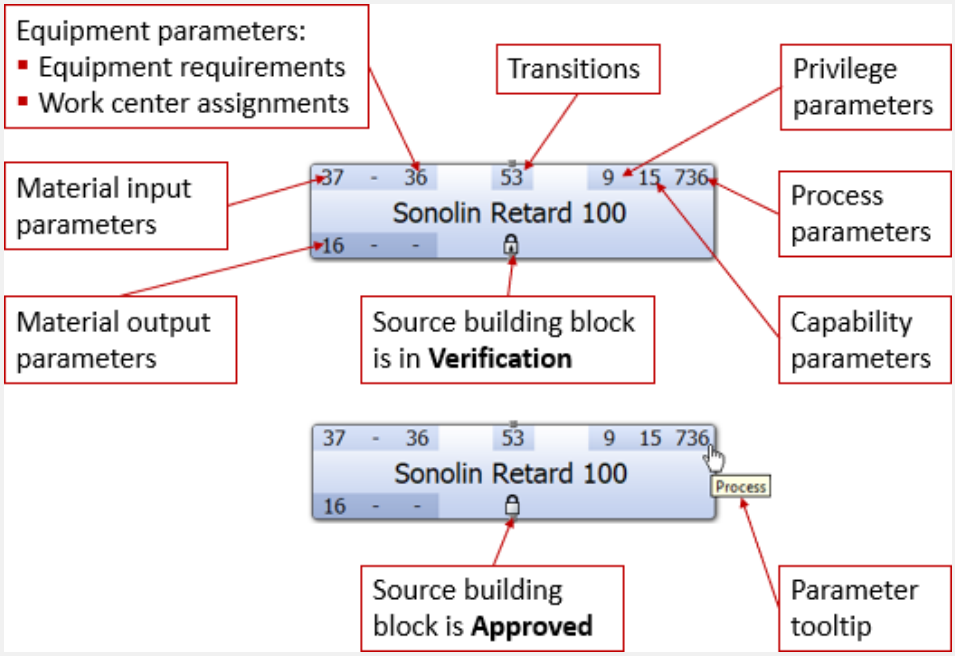


Figure 8: Custom building block image

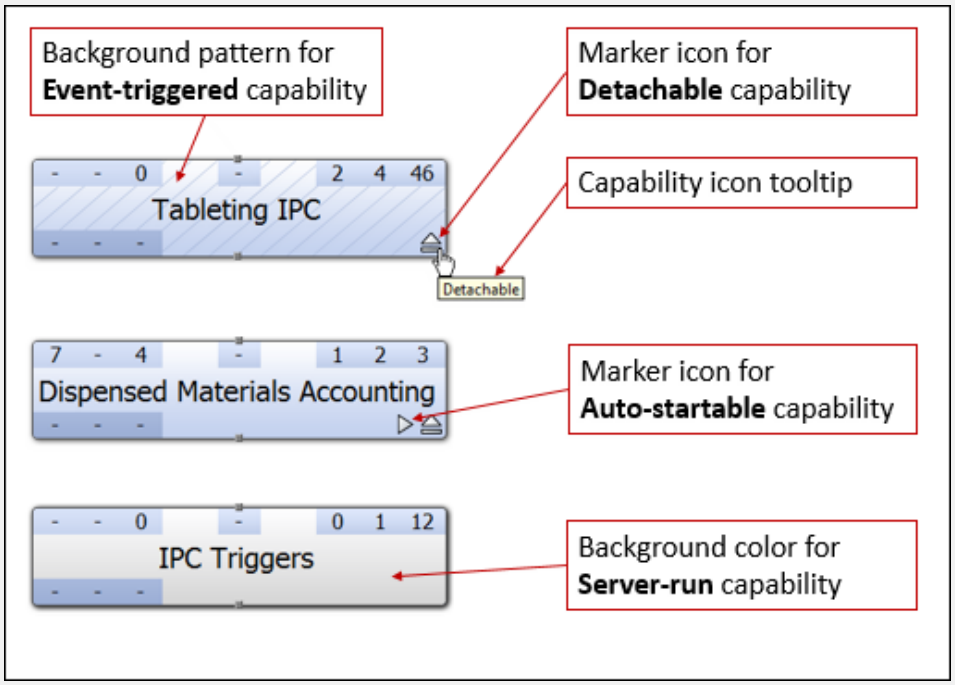


Figure 9: Operation building block image with capabilities

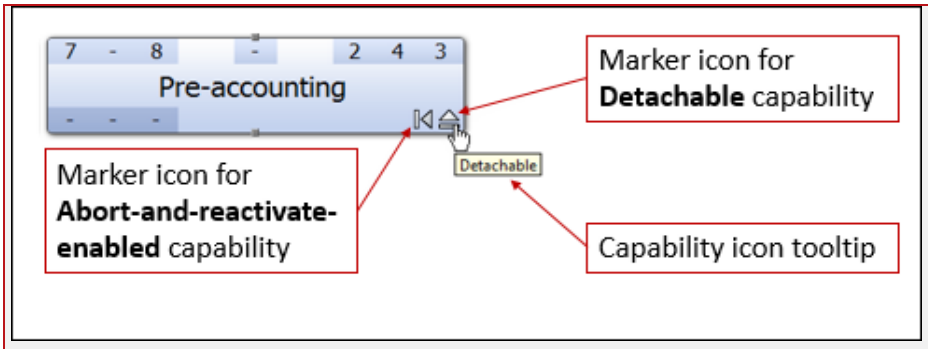


Figure 10: Unit procedure building block image with capabilities

Upper left section:

- Materials as **process input (SR3146.9.5.1)** data (page 18).
- Equipment parameter:
  Equipment requirement (class) (SR3146.9.5.5) data (page 21)
  Work center (SR3146.9.5.4) data (page 21), for unit procedures only
  Station (SR3146.9.5.8) data (page 21), for operations only.

Lower left section:

- Materials as **process output (SR3146.9.6.1)** data (page 27).

Upper right section:

- **Signature privilege (SR3146.9.5.3)** data (page 20).
- Capabilities
  Detachable - Operation (SR3146.9.5.9.1) capability (page 23)
  Event-triggered (SR3146.9.5.9.2) capability (page 24)
  Escalation-enabled (SR3146.9.5.9.3) capability (page 24)
  Trigger-enabled (SR3146.9.5.9.4) capability (page 25)
  Server-run (SR3146.9.5.9.5) capability (page 25)
  Auto-startable (SR3146.9.5.9.6) capability (page 25)
  Pause-enabled (SR3146.9.5.10.1) capability (page 26)
  Detachable - Unit procedure (SR3146.9.5.10.2) capability (page 26).
- Capabilities (Does not apply to Workflow Designer.)
  Abort-and-reactivate-enabled (SR3146.9.5.10.3) capability (page 27).
- **Process parameter (SR3146.9.7+)** data (page 29).

Upper center section:

- **Transition (SR3146.9.5.11)** data (page 23).

Center section:

- Name of the building block.

Lower center section:

- **Frozen (SR3146.9.13.2)** markers (page 32) (if the used custom building block is in the **Verification** or **Approved** statuses).

Process View (SR3146.9.1.1)

A process-related view is available to support the creation and maintenance of procedural elements of the **procedural structure (SR3146.9.4)** hierarchy (page 13).

WORKBENCH - START AND END STEPS (SR3146.9.1.1.1)

On each level of the procedural structure, Start and End steps are available. Thus, a recipe or workflow author can model branches and loops that include the first or final building blocks of a graph.

Examples: A process flow starts with parallel steps. The last step of a process flow can have a loop.

Group of Building Blocks Representation (SR3146.9.1.7)

A building block used as phase or operation group is recognizable as such in the graph and can be selected as one element.

Navigation (SR3146.9.1.2)

The following navigation and correction tools are available:

- Zoom
- Map (bird eye)
- Explorer
- Pan current graph
- Hierarchical navigation
- Horizontal navigation (within one master recipe, master workflow, or custom building block)
- Undo last graph-related user actions (independent of the current Graph Window).
- Redo actions revoked by the **Undo** function.

- Zoom - Support of zoom in, zoom out, zoom to fit, and 100% of current graph.
- Map - Displays the current building block and its superordinate building blocks in three views and enables the navigation through the master recipe or master workflow. The view of the current Graph Window is highlighted. The Map applies to procedure level, unit procedure level, and operation level.
- Explorer - Displays a structural overview of the graph and enables the navigation through the master recipe, master workflow, or building block.
  A corresponding marker indicates that a graph element is based on a source cBB in the **Verification** or **Approved (SR3146.10.1)** statuses (page 87).
- Hierarchical navigation - Go up to superordinate procedural level and go back to previous procedural level.

Context Menus (SR3146.9.1.3)

Context menus enhance the usability during the creation and maintenance of procedural elements of the **procedural structure (SR3146.9.4)** hierarchy (page 13).

Context menus are available for tab titles of the **Graph Window (SR3146.9.2.1)** UI element (page 136), the work area of the Graph Window, the building block images (page 130), the **Graph Clipboard (SR3146.9.1.6.3)** table (page 134) of the Setlist, and the **Material Flow Control (SR3146.9.8)** tab (page 43).

Copy & Paste Function (SR3146.9.1.6+)

When you design the SFC graphs of recipes, workflows, or building blocks in the Graph Window of Recipe Designer or Workflow Designer, the system supports you with a copy and paste function. It allows you to copy a building block (phase, phase group, operation, operation group, unit procedure, or procedure) and paste it at another location in the current graph or into any other graph on the same structure level. The configuration of the building block is fully retained with all defined parameters and sub-graphs.

Copy a Building Block (SR3146.9.1.6.1)

A selected building block in the Graph Window can be copied to the Graph Clipboard. The system copies the selected building block including its defined parameters and sub-graphs. A building block is copied in its current state, even with unsaved changes.

COPY SEVERAL BUILDING BLOCKS (SR3146.9.1.6.4)

Several selected building blocks (phase, phase group, operation, operation group) in the Graph Window can be copied to the Graph Clipboard. Such selected building blocks must fulfill together the **Checks that Apply to Group Modeling** checks set (page 56) and the following further rules.

- All selected building blocks must be connected to each other.
- All selected building blocks must be elements of one group or complete group(s) or only non-group elements or complete group(s) and non-group elements.
- Branches must be selected completely with a building block before as start and a building block after as end element.

The system copies the selected building blocks including their defined parameters and sub-graphs. A building block is copied in its current state, even with unsaved changes.

PASTE A BUILDING BLOCK (SR3146.9.1.6.2)

A building block or building block group that is available in the graph clipboard can be pasted into the current Graph Window tab if its structure level matches.
Adding a building block (group) as an unconnected component to the graph shall be possible.
Adding a building block (group) at a specific position in the graph shall be possible with the Graph Clipboard of the Setlist. When using the Setlist, the insertion rule of the building block can be specified.

SETLIST - GRAPH CLIPBOARD (SR3146.9.1.6.3)

The Graph Clipboard makes copied building blocks available to add them together with the Setlist toolbar. A copied building block is listed with

- its identifier (in case of a group or several building blocks, the identifier of the first building block),
- its structure level (+ **group** in case of a group or several building blocks),
- the time when it was copied, and
- a description (initially empty).

The Graph Clipboard table is sorted by the copy time, so the last copied building blocks are always at the top. To be able to differentiate between building blocks that may have the same identifier, a description of up to 80 characters can be defined.

Recipe Designer and Workflow Designer use a common Graph Clipboard. It shall allow to copy an element in one designer and paste it into the other designer.

The Graph Clipboard is cleared when you log off from Recipe and Workflow Designer.

Graph Clipboard Context Menu (SR3146.9.1.6.3.1)

For a selected building block, the context menu shall offer the possibility to

- rename the identifier,
- edit the description,
- remove it from the Graph Clipboard,
- remove all other building blocks from the Graph Clipboard.

Additionally, the context menu allows to clear the entire Graph Clipboard.

Docking (SR3146.9.1.4)

The docking configuration is available as a default layout and a user-specific layout.

In the default docking configuration, the windows are arranged as follows:

- To the left of the Graph Window: **Map (SR3146.9.1.2)** tool (page 132) and **Explorer (SR3146.9.1.2)** tool (page 132).
- To the right of the Graph Window: **Setlist (SR3146.9.2.9)** UI element (page 140) and **property windows** (SR3146.9.2.12 (page 144), SR3146.9.2.13 (page 144), SR3146.9.2.15 (page 144)).
- At the bottom of the Graph Window: **Phase Preview (SR3146.9.2.11)** UI element (page 148), **Messages (SR3146.9.2.16+)** window (page 144), **Comparison (SR3146.9.2.19+)** window (page 146).

The following functions are available to maintain the docking configuration:

- Undo last layout changes related to the dockable windows.
- Redo last layout changes related to the dockable windows.
- Save the window layout (user-specific).
- Load the last saved window layout (user-specific).
- Reset the window layout to the default layout.

Tooltips (SR3146.9.1.5)

Tooltips are available for the following objects:

- transitions (page 15),
- parameter buttons in the building block images (page 130), and
- **MFC graph (SR3146.9.8.1)** components (page 43).

Workbench UI Elements (SR3146.9.2+)

The graphical Designer workbench provides several UI elements (e.g. Library Universe, Messages window, Parameter Panel).

Work Area (SR3146.9.2.1)

The work area panel consists of the Graph Window.

- The Graph Window can display more than one building block of a master recipe, master workflow, custom building block, or change request, each in a separate tab (subordinate tab).
- Subordinate tabs can be closed separately.
  Does not apply to change requests.
- The tab title of subordinate tabs displays the custom building block identifiers.
  Does not apply to change requests.
- An asterisk in the upper tab title is the indicator for unsaved master recipes, master workflows, custom building blocks, or change requests.
- An R in the upper tab title is the indicator for read-only master recipes, master workflows, custom building blocks, or change requests.
- All tabs can be closed one at a time except for the current master recipe, master workflow, custom building block, or change request.
- The graph and its components can be arranged automatically in order to center sequences and branches with respect to their start components and to optimize the vertical spacing between individual components.
- The graph can be compacted automatically for optimal use of space.
- Grid lines can be shown or hidden.

Open Dialog (SR3146.9.2.20+)

The **Open** dialog is related to **Master recipe (SR3146.9.10.1)** objects (page 33), **Master workflow (SR3146.9.10.9)** objects (page 34), **Custom building block (SR3146.9.10.7)** objects (page 84), and **Change request (SR3146.12+)** objects (page 97), and system building blocks. It allows to select data records for opening in a resizable dialog. It provides an intelligent filter and search function.

- A quick search function applies to the search result and reduces the result according to the filter criteria.
- A filter provides various hierarchy levels. The selection of an item on a hierarchy level reduces the subsequent search results.
- The search result is available via a separate table.

OPEN DIALOG - HIDE OBJECTS (SR3146.9.2.20.2)

By default, master recipes and master workflows in a **Retired** status (e.g. **Archived**, **Obsolete**) and procedural elements in the **Archived** status are excluded from the search result.

The inclusion of master recipes and master workflows in a **Retired** status and the inclusion of procedural elements in the **Archived** status can be switched on.

OPEN DIALOG - MISSING REFERENCED OBJECT (SR3146.9.2.20.3)

As long as an object is in an editable status, objects it references from its parameters can be deleted, e.g. an equipment requirement references an equipment class that is deleted.

Nevertheless, it shall still be possible to open the object in the editable status. Missing references are automatically removed and an information message is displayed in this regard.

OPEN DIALOG FOR PROTECTED OBJECTS (SR3146.9.2.20.1)

For objects that are protected by an access privilege: The system shall only provide objects for selection whose access privilege matches the access privilege of the logged-in user.

Library Universe (SR3146.9.2.4+)

The Universe works as a library and allows to select data records for further processing in a resizable dialog.

The same library style is used for the following actions:

- Select a **material (SR3146.9.2.4.5)** data record (page 138) during the creation of a new master recipe.
- Select a **material (ERP BOM) (SR3146.9.2.4.2)** data record (page 139) during the creation of a new master recipe.
- Open a master recipe, master workflow, or custom building block by means of the **Open (SR3146.9.2.20+)** dialog (page 136).

Library Filter & Search (SR3146.9.2.4.3)

The Universe provides an intelligent filter and search function.

- A quick search function applies to the search result and reduces the result according to the filter criteria.
- A filter provides various hierarchy levels. The selection of an item on a hierarchy level reduces the subsequent search results.
- The search result is available via a separate table.
- Each item of the search result can be flagged as Setlist item in order to be available in the **Setlist (SR3146.9.2.9)** UI element (page 140).

Library for Building Blocks (SR3146.9.2.4.4)

The Universe provides building block-specific libraries to select procedures, unit procedures, operations, and phases.

For building block objects that are protected by an access privilege: The system shall only provide building block objects whose access privilege matches the access privilege of the logged-in user.

The library for building blocks on the operation and unit procedure levels allows to apply a filter on the **Usage as group** attribute.

Library for Materials (SR3146.9.2.4.5)

The Universe provides a material-specific library to select corresponding data records. The list of materials can be filtered per material type.

Library for Materials (ERP BOM) (SR3146.9.2.4.2)

➢ Does not apply to Workflow Designer.

The workbench provides a material (ERP BOM)-specific library to select a product material for which an ERP BOM exists.

The library is only available during recipe creation in order to assign the product material to the recipe.

- For ERP BOMs that are protected by an access privilege: The system shall only allow to select product materials whose access privilege matches the access privilege of the logged-in user.
  The access privilege of the ERP BOM is passed on to the master recipe. In the default configuration, the access privilege cannot be changed.
- For ERP BOMs that are not protected by an access privilege: The system shall allow to select an access privilege for the protection of the master recipe from unauthorized access for users with access privileges that allow to maintain protected recipes. The list of available access privileges shall be restricted to the access privileges of the logged-in user.

The library is not available in the **Universe (SR3146.9.2.4+)** UI element (page 137).

Library for Work Centers (SR3146.9.2.4.1)

The Universe provides a work center-specific library to select corresponding data records.

Library for Stations (SR3146.9.2.4.9)

The Universe provides a station-specific library to select corresponding data records.

Library for Equipment Classes (SR3146.9.2.4.7)

The Universe provides an equipment class-specific library to select corresponding data records. The list of equipment classes can be filtered per equipment class level.

Library for Property Types (SR3146.9.2.4.8)

The Universe provides a property type-specific library to select corresponding data records. The list of property types can be filtered per data type.

Library for Signature Privileges (SR3146.9.2.4.6)

The Universe provides a signature privilege-specific library to select corresponding data records.

Setlist (SR3146.9.2.9)

A Setlist provides pre-selected data records for building master recipes, master workflows, custom building blocks, and change requests.

- The data records (e.g. material, building blocks) are pre-selected in the **Universe (SR3146.9.2.4+)** UI element (page 137).
- Capabilities are provided by the system.
- The Setlist is easily accessible during recipe, workflow, custom BB, and change requests building.
- Setlist items are added to the respective section of the recipe, workflow, and custom BB with a mouse click.
- BBs are added to a change request with a mouse click.
- ERP BOM items are only available within Recipe Designer.

The Setlist provides functions for

- building sequences, selection branches, simultaneous branches, and joining elements according to the **procedural structure (SR3146.9.4)** hierarchy (page 13),
- inverting the default inserting direction,
- (smart) replacing (system) phases, operations, unit procedure, procedures,
- assigning materials and ERP BOM items as **process input (SR3146.9.5.1)** parameters (page 18) and **process output (SR3146.9.6.1)** parameters (page 27) to phases,
- assigning **work center (SR3146.9.5.4)** parameters (page 21) to unit procedures,
- assigning **station (SR3146.9.5.8)** parameters (page 21) to operations,
- assigning **equipment requirement (class) (SR3146.9.5.5)** parameters (page 21) as process input to phases,
- assigning **signature privilege (SR3146.9.5.3)** parameters (page 20) as process inputs to phases, and
- assigning **capability (SR3146.9.5.9.)** parameters (page 23) as process inputs to operations or unit procedures.

ERP BOM items are only available for master recipes based on an ERP BOM; in the **ERP BOM Item (SR3146.9.2.9.1)** section (page 141) of the Setlist.

SETLIST - ERP BOM ITEM (SR3146.9.2.9.1)

➢ Does not apply to Workflow Designer.

The **ERP BOM Item** section is filled automatically with BOM items (and their common attributes) of the ERP BOM on which a master recipe is based.
When an ERP BOM item is assigned to a recipe/building block element, the material data and common BOM item attributes are populated into the corresponding material input parameter.
Common attributes are position, material identifier and short description, quantity with UoM, and fixed quantity.

SETLIST - WORKFLOW-RELATED CONSISTENCY CHECKS (SR3146.9.2.9.2)

In Workflow Designer, the Setlist can hold custom building blocks that are not conform to the required master workflow structure.

- If the conflict cannot be solved within the master workflow, the system does not allow to add the custom building block to a master workflow (e.g.: a custom procedure holds more than one unit procedure).
- If the conflict can be solved within the master workflow, the system allows to add the custom building block to the master workflow and displays a corresponding error message in the **Messages (SR3146.9.2.16+)** window (page 144) (e.g. a custom operation holds a phase that is not marked for usage within a workflow).

SETLIST - WORKFLOW-RELATED DATA ADJUSTMENT (SR3146.9.2.9.3)

In Workflow Designer, the Setlist can hold custom building blocks with recipe-related attribute definitions. This applies to:

- **Dispense** attribute for operations/unit procedures (**SR3146.9.4.4** operation type (page 16)).

If a custom building block is added to a master workflow, the recipe-related attribute definitions are ignored and removed. (They bear no meaning for workflows and cannot be edited in the context of a workflow.)

SETLIST - GROUP OF BUILDING BLOCKS (SR3146.9.2.9.4)

Operations and unit procedures can be used as a groups of building blocks if their group usage attribute is set to **Allowed** or **Mandatory**. For ease of use of such elements, separated sections are available in the Setlist:

- **Operation Group**
- **Phase Group**

If a unit procedure with the group usage attribute **Allowed** is added to the Setlist, the unit procedure is added to both the **Unit Procedure** and **Operation Group** sections.
If a unit procedure with the group usage attribute **Mandatory** is added to the Setlist, the unit procedure is added to the **Operation Group** section only.
If a unit procedure with the group usage attribute **Not allowed** is added to the Setlist, the unit procedure is added to the **Unit Procedure** section only.
If an operation with the group usage attribute **Allowed** is added to the Setlist, the operation is added to both the **Operation** and **Phase Group** sections.
If an operation with the group usage attribute **Mandatory** is added to the Setlist, the operation is added to the **Phase Group** section only
If an operation with the group usage attribute **Not allowed** is added to the Setlist, the operation is added to the **Operation** section only.

If the group usage attribute of an operation or unit procedure is changed and it has already been added to the Setlist before, the sections are updated accordingly.

Parameter Panel (SR3146.9.2.7)

A Parameter Panel allows to maintain existing parameters, transitions (with an identifier), and their attributes.

- If applicable, parameters can be deleted.
- Parameters and transitions can be locked and unlocked.
  - Locking of parameters (SR3146.9.13+) feature (page 31)
  - Locking of transitions (SR3146.9.14+) feature (page 32)
- The **Lock** attribute provides the following information:
  - Frozen: source cBB is in the **Verification** or **Approved** statuses.
  - Locked: parameter/transition is locked.
  - Unlocked: parameter/transition is unlocked.
- Allows to filter per parameter type (e.g. process parameter, material parameter, transition).

- The Parameter Panel is resizable.
- Applies to material parameters:
  In its vertical pane, the Parameter Panel displays each attribute of a parameter
  In its horizontal pane, the Parameter Panel displays only the **Basic** attributes of a parameter.

Default attributes for each parameter are:

- Lock
- Hierarchy level
- Path

Parameters and transitions (with identifiers) are accessible via buttons on the building block images (page 130).

Cell editors facilitate easy editing of values: Measured Value editor (page 158), Multi-line text editor (page 158), **Expression editor (SR3146.9.9.4)** UI element (page 148), **Date/Time Picker (SR3146.9.2.17.1)** editor (page 159), **Duration (SR3146.9.2.17.2)** editor (page 159), **Packaging Level Data (SR3146.9.2.17.3)** editor (page 159), and further editors (page 160).

Materials, signature privileges, work center parameters, equipment requirement parameters, property types, capabilities, and dynamically created process parameters can be deleted as long as they are not locked by means of the **Lock (SR3146.9.13+)** attribute (page 31).

Applies to material parameters:
The vertical pane of the Parameter Panel is collapsible.

Parameter Panel - View All (SR3146.9.2.7.1)

The Parameter Panel can be accessed from all levels of the S88 building blocks. The visible parameters include all of the local parameters of the building block below the current building block level (view all per building block).

Parameter Panel - Specific Property Types (SR3146.9.2.7.2)

Process input parameters can be refined by adding one or more parameter-specific property types and, if applicable, by defining a specific content for each property type.

This capability applies to equipment requirement parameters.

PARAMETER PANEL FILTER & SEARCH (SR3146.9.2.8)

The Parameter Panel provides an intelligent filter and search function.

- A quick search function applies to the search result and reduces the result according to the filter criteria.
- A filter provides various hierarchy levels. The selection of an item on a hierarchy level reduces the subsequent search results.
- The search result is available via a separate table.

Property Windows

The workbench provides property windows specific to master recipe, master workflow, building block, and change request components.

Cell editors facilitate easy editing of values: Measured Value editor (page 158), Multi-line text editor (page 158), and **Packaging Level Data (SR3146.9.2.17.3)** editor (page 159).

Property window - Header (SR3146.9.2.15)

Provides the properties of the header component (master recipe, master workflow, custom building block, change request).

Property window - Element (SR3146.9.2.12)

Provides the properties of the element component within a master recipe, master workflow, or custom building block.

Property window - Source BB (SR3146.9.2.13)

Provides the properties of the source system/custom/group building block that was used as a template.
This applies to the usage of a BB as a master recipe, master workflow, or building block element, "Save as" of a BB, and "Create BB from selected".

Messages (SR3146.9.2.16+)

The **Messages** window displays messages of various categories related to the current master recipe, master workflow, or building block. They result from checks performed by the Designer workbench when master recipes, master workflows, or building blocks are being edited or imported.

User Interface of the Messages Window

The tool panel displays the number of messages per message category and a breadcrumb trail that indicates the path to the element that is currently selected in the tree panel.

The tree panel displays a structural overview with all nodes of the master recipe, master workflow, or building block for which the checks have found messages. Navigating the tree highlights messages in the list panel and updates the breadcrumb trail.

The list panel displays a list of all messages that have been found for the master recipe, master workflow, or building block. Messages are prepended by unique message/message category identifiers.

Navigating the list highlights nodes in the tree panel, updates the breadcrumb trail, can select affected elements in the Graph Window, can select affected parameters in the Parameter Panel, can select the Material Flow Control tab, can select affected properties in the property windows, and can select expressions in the Expression editor of affected transitions, process parameter attributes, equipment requirement rules, if applicable.

The message category is indicated by an icon in the tool and the list panel.

The tab title also contains the number of messages per message category.

Messages - Validator-related (SR3146.9.2.16.1)

The messages are related to an ongoing validation (consistency checks) of the procedural structure and its parameters. The following message categories are supported:

- Error
  Errors need to be resolved for approved custom building blocks and valid master recipes or master workflows.
- Warning
  Warnings need to be resolved for valid master recipes or master workflows.
- Information
  Validator result that does not require to be resolved.

For consistency check-related messages, see "Validator Consistency Checks (SR3146.9.4.3+)" (page 54).

Messages - Category Filter (SR3146.9.2.16.2)

The display of messages can be filtered by message category.

Messages - Navigation (SR3146.9.2.16.3)

The structural overview in the tree panel allows the navigation based on the tree structure of the master recipe, master workflow, or building block.

MESSAGES - RESOLVE ISSUE (SR3146.9.2.16.4)

For messages caused by inconsistencies that can be resolved automatically, the system provides the **Resolve issue** functions in the context menu of the related message to resolve the selected issue, to resolve all issues of the same message category identifier, or to resolve all issues.

The system allows to resolve a single issue or all issues at once.

The **Resolve issue** function is available for the following checks and warnings:

- **Validator - Phase parameter - ERP BOM attribute discrepancy (SR3146.9.4.3.56)** check (page 64)
- **Validator - Recipe - Planned quantity does not fit ERP BOM (SR3146.9.4.3.57)** check (page 63)
- **Validator - Recipe - Packaging Level Does Not Fit ERP BOM (SR3146.9.4.3.58)** check (page 63)
- **Validator - Material Parameter - Packaging Level Does Not Fit Material's Master Data (SR3146.9.4.3.109)** check (page 69)
- **Validator - Transition condition - Broken link - Error (SR3146.9.4.3.37.2)** check (page 58)
- **Validator - Phase parameter - Broken link - Error (SR3146.9.4.3.25.2)** check (page 58)
- **Validator - Equipment requirement rule Broken link - Error (SR3146.9.4.3.90.2)** check (page 77)
- **Import Consistency Check Messages (SR1075.1.3.5.5+)** warnings (page 170)

Comparison (SR3146.9.2.19+)

The **Comparison** window displays the result of a comparison of the current master recipe, master workflow, or custom building block with the comparison baseline. See also "Comparing Master Recipes, Master Workflows, and Custom Building Blocks (SR3146.15+)" (page 93).

User Interface of the Comparison Window

The tool panel displays the number of differences per difference category, an option list to define the objects whose differences are displayed, and a breadcrumb trail that indicates the path to the element that is currently selected in the tree panel.

The tree panel displays a structural overview with all nodes of the master recipe, master workflow, or custom building block for which the comparison has found differences. Navigating the tree highlights entries in the list panel and updates the breadcrumb trail.

The list panel displays a list of all differences that have been found for the master recipe, master workflow, or custom building block.

Navigating the list highlights nodes in the tree panel, updates the breadcrumb trail, can select affected elements in the Graph Window, can select affected parameters in the Parameter Panel, can select the Material Flow Control tab, can select affected properties in the property windows, and can select expressions in the Expression editor of affected transitions, process parameter attributes, equipment requirement rules, if applicable.

The difference category is indicated by an icon in the tool and the list panel.

The tab title also contains the number of differences per difference category.

Removed objects are displayed last under their parent nodes.

Comparison - Difference Categories (SR3146.9.2.19.1)

The following difference categories are supported:

- Added
  A parameter, parameter's child node (e.g. action of a capability), or procedural element has been added.
- Changed
  An attribute, parameter, parameter's child node, procedural element, MFC-related setting, or the layout has been changed.
- Removed
  A parameter, parameter's child node, or procedural element has been removed.

For the difference-specific handling, see "Comparison Algorithm (SR3146.15.1+)" (page 93).

Comparison - Category Filter (SR3146.9.2.19.2)

The display of differences can be filtered by difference category.

Comparison - Type Filter (SR3146.9.2.19.3)

The display of messages can be filtered by type.

COMPARISON - NAVIGATION (SR3146.9.2.19.4)

The structural overview in the tree panel allows the navigation based on the tree structure of the master recipe, master workflow, or custom building block.

COMPARISON - DIFFERENCE-SPECIFIC INFORMATION (SR3146.9.2.19.5)

The following difference-specific information is available:

- Type
- Identifier
- Predecessor
- Successor
- Attribute
- New value
- Old value

Preview Panel (SR3146.9.2.11)

The Preview panel consists of the Phase Preview which displays how the selected phase will be rendered when executed.

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

DEFAULT PREVIEW BUILDTIME (SR3146.9.9.4.1)

If a phase returns a preview-related error, the preview of the phase in the **Preview Panel (SR3146.9.2.11)** UI element (page 148) is replaced by a default preview.
Example: Input parameters are not available.

Expression Editor (SR3146.9.9.4)

The Expression editor supports:

- References to output variables (e.g. used for information flow between phases and operations)
- References to attributes of equipment properties (e.g. used for definition of equipment-related rules)
- Operators (see SR3146.9.9.4.4 (page 151))
- Functions (see SR3146.9.9.4.9 (page 156), SR3146.9.9.4.7 (page 152), SR3146.9.9.4.11 (page 153), and SR3146.9.9.6 (page 153))

The editor is available for **process parameter (SR3146.9.7+)** data (page 29), **rule (SR3146.9.5.7)** definitions (page 22), and **transition condition (SR3146.9.4.2)** definitions (page 16).

Intelligent auto-completion feature supports the selection of expression components.

The list of available expression elements displays indicators for specific elements with details as tooltip (e.g. parallel elements, event-triggered elements, successor elements).

Transition identifier of an expression is displayed within the transition in the Graph Window and as part of the respective tooltip.

Rule identifier of an expression is displayed within the process input requirement identifier.

Textual description of an expression is displayed as value in the parameter panel and as part of the tooltip for the transition in the Graph Window, respectively.

Concurrent syntax and semantics validation supports the definition of expressions and references.

Level indicators are supported in the Expression editor for the references of unit procedures (UP:), operations (OP:), and phases (PH:) procedural elements.

Output references consist at least of a procedural element identifier and an output identifier. For references across higher level procedural elements, their identifiers are also components of the output reference.

Blanks, tabs, and line breaks can be used for better readability.

Round brackets can be used to override or clarify an operators' precedence.

Single-line and multi-line comments can be added.

Indicators for errors (red underline of expression, error icon at affected line, global error status marker, error message as tooltip).

Code folding is supported to enable a clear structuring of large expressions (except for **Pnuts (SR3146.9.9.4.6)** scripting (page 156)).

Expressions for Transitions (SR3146.9.9.4.2)

Conditions within a transition can be defined with the Expression editor.

In case a condition or a description is defined, the definition of a transition identifier is mandatory. Transition identifiers are generated by PharmaSuite and can be edited by the recipe or workflow author. They must be unique within their procedural element of the **procedural structure (SR3146.9.4)** hierarchy (page 13).
For transitions that only hold the default condition, identifiers are optional.

User-defined transition conditions apply in addition to the default condition of the **transition (SR3146.9.4.2)** condition (page 16).
The expression result must be of the **Boolean** type.

Default attributes for each transition are:

- Identifier
- **Lock (SR3146.9.14+)** attribute (page 32)
- Description
- Expression

Expressions for Process Parameters (SR3146.9.9.4.3)

Process parameters can be defined with the Expression editor.

An editor-specific icon indicates the editor's availability.
The data type of the expression result must match the attribute's data type.

Expressions for Flexible Rules (SR3146.9.9.4.10)

Flexible rules within a process input requirement can be defined with the Expression editor.

In case a flexible rule is added to a process input requirement, the definition of a rule identifier is mandatory. Rule identifiers are generated by PharmaSuite and can be edited by the recipe or workflow author. They must be unique within their phase.

The expression result must be of the **Boolean** type.

Expressions for Conditional Rules (SR3146.9.9.4.12)

Conditional rules within a process input requirement can be defined with the Expression editor.

They are group-enabled, i.e. per default they apply as an equipment requirement to the identified parent entity and each of its child entities.

The Expression editor supports the following keywords to define a conditional rule:

- **provided that** allows to define conditions that must be fulfilled and
- **require** allows to define the checks themselves.

If the condition of the **provided that** section is not fulfilled for at least one entity of the group, the conditional rule fails.

In case a conditional rule is added to a process input requirement, the definition of a rule identifier is mandatory. Rule identifiers are generated by PharmaSuite and can be edited by the recipe or workflow author. They must be unique within their phase.

The expression result must be of the **Boolean** type.

Expression Editor Operations (SR3146.9.9.4.4)

The Operators tree panel provides the following operators:

- Arithmetic operators: + , - , *, /
- Comparison operators: ==, !=, <, <=, >, >=
- Logical operators: AND, OR, NOT
- Unary arithmetic operators: + , -
- String concatenation: +

A special handling applies to the Date and Duration data types:

- Date - Date = Duration
- Date (+, -) Duration = Date
- Duration (*, /) Scalar = Duration

A special handling applies to the Measured value (MV) data type:

- MV (*, /) Scalar = MV
- MV (+, -) MV = MV
- MV (/) MV = Scalar.

Expression Editor Literals (SR3146.9.9.4.5)

The following literals are supported:

- Long - integral numbers: e.g. 12345
- Float - floating point numbers (fractional numbers): e.g. 123.45Float
- BigDecimal - floating point numbers (fractional and integral numbers) with greater precision than Float: e.g. 123.45 or 123BDecimal or 12.3e42
- Boolean: true, false
- String: e.g. "This is a string."
- Date and time: e.g. 07/19/2012 9:20:00 AM CEST
- Duration: e.g. 23d1h44min3s
- Measured value: e.g. 23kg

Additional literals:

- IMESS88Equipment: equipment data object
- PhaseDataReference collected phase data

Expression Editor Functions (SR3146.9.9.4.7)

The Functions tree panel provides the following functions:

- Current Date and Time function: now()
- Convert to Unitless Number function: convertTo(arg1, arg2)
- Convert to BigDecimal function: convertToBigDecimal(arg1)
- Convert to Long function: convertToLong(arg1, arg2)
- Convert to MeasuredValue function: convertToMeasuredValue(arg1, [arg2])
  Use this function for units of measure that cannot be interpreted correctly due to the usage of special characters (e.g. °C or IU/ml).
- Convert to String for Display function: convertToDisplayString(arg1)
- Convert for Date Only Use: convertForDateOnlyUse(arg1)
  Use this function to convert a timestamp for use as date only (time set to midnight).
- String Contains function: containsItem(arg1, arg2, [arg3])
- Average function: average(arg1, arg2, ...)
- Minimum function: min(arg1, arg2, ...)
- Maximum function: max(arg1, arg2, ...)

- Null-handling functions: see SR3146.9.9.4.9 (page 156)
- Phase-specific functions: see SR3146.9.9.4.11 (page 153)
- Runtime context data functions: see SR3146.9.9.6 (page 153)
- Average function is only supported by Numeric data types.

EXPRESSION EDITOR - PHASE-SPECIFIC FUNCTIONS (SR3146.9.9.4.11)

The Functions tree panel provides the following function only if it is supported in the context of the specific phase (e.g. **Identify equipment** phase (SR0300+)):

- Reference to the currently identified equipment: currentEquipmentEntity (equipment context free)

EXPRESSION EDITOR - RUNTIME CONTEXT DATA (SR3146.9.9.6)

The Functions tree panel provides the following functions related to runtime context data:

- General context
  - Create Workflow ID to generate a unique workflow identifier: createWorkflowId(arg1)
    with the prefix to use as argument.
  - Get Context Data Value to retrieve the context value for a key.
    The context is written by the phase **Write context data** phase:
    getContextValue(arg1, arg2, arg3)
    - with the data type of the value to retrieve as first argument,
    - with the key of the value to retrieve as second argument,
    - with the context ID of the value to retrieve as optional third argument. If it is not provided the current order/workflow ID is used.
  - Does not apply to Recipe Designer
    - Workflow Is Appended to Order to check if the current workflow is appended to any order: workflowIsAppendedToOrder()
- Order context
  - Order/Workflow ID function: orderId
  - Does not apply to Workflow Designer.
    - Order Scaling Factor function: orderScalingFactor
    - Produced Material ID function: producedMaterialId
    - Produced Material Short Description function: producedMaterialShortDescription

- Produced Material Type function: producedMaterialType
- Batch ID function: batchId
- Batch Status function: batchStatus

- MFC position context

  The functions of the MFC position context are only supported for Dispense, Inline Weighing, and Output Weighing; they are not supported by the Material Tracking phases (**Identify material**, **Produce material**, **Account material**).

  - Material ID function: materialId
  - Material Short Description function: materialShortDescription
  - Material Type function: materialType
  - Status function: mfcPositionStatus
  - Planned Quantity - Original function: originalPlannedQuantity
  - Planned Quantity - Execution function: plannedQuantity
  - Actual Quantity - Last split function: actualSplitQuantity
    Returns the actual recorded quantity of all the last split positions that constituted the last target of a material input position. For material outputs, the function always returns null.
  - Actual Quantity - Total function: totalActualQuantity
    Returns the total recorded quantity of a material position minus replaced quantity or quantity that was declared as waste.
  - Sum of Actual Input Quantities - Total function: sumTotalActualInputQuantity
    Returns the total recorded quantity of all material input positions of the unit procedure or the specified material input positions of the procedure.
  - Prorate Factor function: prorateFactor
    Does not apply to Workflow Designer.
  - Yield function: yield
    Does not apply to Workflow Designer.
  - Production Date - Last Identified Sublot function: productionDateOfSublot
    Returns the production date of the last identified sublot of a material input position. In case no sublot has been identified or for material outputs, the function always returns null.

- Equipment context
  - Equipment Is Bound to a unit procedure: equipmentIsBound(arg1)
    with an equipment identifier as argument.
  - Current equipment entity (runtime-related) function: currentEquipmentEntity

- Equipment ID to retrieve the identifier of an equipment entity: equipmentId(arg1)
  with an equipment object as argument.
- Equipment is member of class function: equipmentIsMemberOfClass(arg1, arg2)
  With an equipment object as first argument and an equipment class identifier as second argument.
- Equipment property attribute (runtime-related) function: equipmentPropertyAttribute(arg1, arg2 [, arg3, arg4])
  With an equipment object as first argument, a property type identifier as second argument, a property type attribute identifier as optional third argument, and an index as optional fourth argument.
- Equipment has property attribute function: equipmentHasPropertyAttribute(arg1, arg2 [, arg3, arg4])
  With an equipment object as first argument, a property type identifier as second argument, a property type attribute identifier as optional third argument, and an index as optional fourth argument.
- Equipment graph status (runtime-related) function: equipmentGraphStatus(arg1, arg2 [, arg3])
  With an equipment object as first argument, a graph purpose as second argument, and an attribute identifier (KEY or DISPLAY) as optional third argument.
- Equipment has graph function: equipmentHasGraph(arg1, arg2)
  With an equipment object as first argument and a graph purpose as second argument.
- Equipment graph expiry date (runtime-related) function: equipmentGraphExpiryDate(arg1, arg2)
  With an equipment object as first argument and a graph purpose as second argument.
- Equipment FSM status function: equipmentFSMStatus(arg1, arg2 [, arg3])
  With an equipment object as first argument, the identifier of a FlexibleStateModel property type as second argument, and as optional third argument either the "KEY" string or the "DISPLAY" string.
- Date and time of last equipment status change (logbook-related) function: lastEqTransitionTimestamp(arg1, arg2, arg3[, arg4])
  With an equipment object as first argument, the identifier of a FlexibleStateModel property type as second argument, the non-localized state name or the localized status of the entity as third argument, and as optional fourth argument either the "KEY" string or the "DISPLAY" string.
- Last product ID (logbook-related) function: lastProduct

- Location context
  - Work Center ID function: workCenterId
  - Work Center Description function: workCenterDescription
  - Station ID function: stationId
  - Room ID function: roomId
  - Room Status function: roomStatus
  - Storage Area ID function: storageAreaId
  - Storage Area Description function: storageAreaDescription

EXPRESSION EDITOR - IMPLICIT NULL HANDLING (SR3146.9.9.4.8)

The Expression editor implicitly handles NULL markers according to the SQL standard. This leads to the following behavior for expressions including NULL:

- Mathematical expressions result in NULL
  Example: 3 + NULL = NULL
- Aggregation functions eliminate NULL
  Example: MIN(3,NULL,5) = 3
- Comparative expressions result in NULL
  Example: 4 > NULL results in NULL
- Boolean expressions follow the Three-valued logic (3VL).

For transitions, NULL is mapped to FALSE.

EXPRESSION EDITOR - EXPLICIT NULL HANDLING (SR3146.9.9.4.9)

The Functions tree panel provides the following functions for the explicit handling of NULL markers:

- Is undefined: isNull(arg1)
- Undefined - one of two: nvl(arg1, arg2)
- Undefined - one of many: coalesce(arg1, arg2, ...)

EXPRESSION EDITOR - PNUTS (SR3146.9.9.4.6)

The Expression editor provides Pnuts scripting.

The Pnuts mode does not support extensive syntax and semantics validation, access to the Operators and Functions tree panels, and code folding.

Expression Editor - Expression Simulation (SR3146.9.9.4.13)

The Expression editor provides a simulation dialog for expressions defined for process parameters, transitions, and flexible or conditional rules.

- The simulation of expressions is not available as long as the expression contains an error.
- For expression elements (output references and functions annotated as non-deterministic) that cannot be evaluated automatically at the time of simulation, the user needs to provide a value. The expression element and the data type required for the value are displayed. It is possible to define NULL (Undefined) as value, if this is a possible value for the expression element.
- Expression elements with a missing or invalid value are marked with an error icon.
- To enter values, data type-dependent editors are used. Only values for data types that are listed in the **Expression Editor Literals (SR3146.9.9.4.5)** section (page 152) and that provide an editor can be entered.
- The expression can be evaluated as soon as all expression elements hold a value.
- The expression result is displayed.
- The data type of the expression result is displayed.

TIP

Functions are programmatically annotated as deterministic or not. This is important if new functions are made available. See section “Configuring the Expression Editor of Recipe and Workflow Designer and Data Manager” in Volume 3 of the Technical Manual “Configuration and Extension” [A5] (page 171).

Other Editors (SR3146.9.2.17+)

The workbench provides various editors that support data entry. This applies to the various types of input data.

Measured Value Editor

The system provides a Measured Value editor for entering a value with a unit of measure.

- An editor-specific icon indicates the editor's availability.
- The editor is available in the **Parameter Panel (SR3146.9.2.7)** UI element (page 142) and the property windows (page 144).
- See also **Measured Value Technical Property Type (SR3071.8.7.14)** in "Functional Requirement Specification Data Management" [A1] (page 171).

Multi-line Text Editor

The system provides a Multi-line text editor for writing text.

- The editor can interpret HTML syntax.
  The system supports the following HTML tags: `<b>`, `<br>`, `<font>`, `<u>`, `<i>`, `<sup>`, `<sub>`, `<ol>`, `<ul>`, and `<li>`.
  The usage of HTML tag attributes is not recommended. For example, for the `<font>` tag, PharmaSuite only supports the `size` and `color` attributes while the `face` attribute to define the usage of a different font family is not supported.
  Attributes are interpreted for the display of text in the Production Execution Client, in reports, and by the export for archive functionality (to PDF/A).
  Text that is formatted with HTML is displayed as plain text including the HTML tags in master recipe reports or master workflow reports. In batch reports and workflow reports, however, formatting with the supported tags is interpreted and applied.
  For plain text, the system can check during execution in the Production Execution Client whether there is sufficient space to display the text in one line. If not, it adds line breaks as needed. If HTML is used, the system cannot determine the required length and the user of the editor has to add line breaks as required.
- The exception text attribute of a process parameter supports only the <b> and
 HTML tags.
- HTML syntax usage is only expected in process parameter attributes for instruction text, instruction table text, and instruction link text attributes. Further usage may have undocumented limitations.
- An editor-specific icon indicates the editor's availability.
- The editor is available in the **Parameter Panel (SR3146.9.2.7)** UI element (page 142) and the property windows (page 144).

Date/Time Picker Editor (SR3146.9.2.17.1)

The system provides a Date/Time Picker editor for entering a date and time value.

- An editor-specific icon indicates the editor's availability.
- The editor is available in the **Parameter Panel (SR3146.9.2.7)** UI element (page 142) and the Change status dialogs.

Duration Editor (SR3146.9.2.17.2)

The system provides a Duration editor for entering a duration value.

- An editor-specific icon indicates the editor's availability.
- The editor is available in the **Parameter Panel (SR3146.9.2.7)** UI element (page 142) and the property windows.
- See also **Duration Technical Property Type (SR3071.8.7.16)** in "Functional Requirement Specification Data Management" [A1] (page 171).

Packaging Level Data Editor (SR3146.9.2.17.3)

The system provides a Packaging Level Data editor for entering the packaging level data: Meaning, Contained number, Inventory level, and a Hide during execution option.

- An editor-specific icon indicates the editor's availability.
- The editor is available in the **Parameter Panel (SR3146.9.2.7)** UI element (page 142) and the **Property Window - Header (SR3146.9.2.15)** UI element (page 144).

List Selection Editor (SR3146.9.2.17.4)

The system provides a List Selection editor for choosing a list of items, such as a set of exception or comment texts.

- An editor-specific icon indicates the editor's availability.
- The editor is available in the **Property Window - Header (SR3146.9.2.15)** UI element (page 144) and the **Property Window - Element (SR3146.9.2.12)** UI element (page 144).

Unit Procedure and Operation Path Editor (SR3146.9.2.17.5)

The system provides an editor for choosing a path that is built from a unit procedure or unit procedure / operation of the current object.

- An editor-specific icon indicates the editor's availability.
- Intelligent auto-completion feature supports the selection of an existing unit procedures and its operations.
- The editor is available in the **Parameter Panel (SR3146.9.2.7)** UI element (page 142).

Further Editors

The editors listed in this section are specified in the "Functional Requirement Specification Data Management" [A1] (page 171).

- BigDecimal editor
  (see **BigDecimal Technical Property Type (SR3071.8.7.3)**)
- Boolean editor
  (see **Boolean Technical Property Type (SR3071.8.7.4)**)
- Expression editor
  (see **Expression Editor (Data Manager) (SR3071.8.1.4)**)
- FlexibleAttributeDefinition editor
  (see **Flexible Attribute Definition Technical Property Type (SR3071.8.7.9)**)
- FlexibleStateModel editor
  (see **Status Model Technical Property Type (SR3071.8.7.6)**)
- FlexibleTagDefinition editor
  (see **Flexible Tag Definition Technical Property Type (SR3071.8.7.8)**)
- Label Layout Selection editor
  (see **Label Layout Selection Editor (SR3071.8.8.1)**)
- String editor
  (see **String Technical Property Type (SR3071.8.7.5)**)

Statistics (SR3146.9.2.18)

The system provides a statistical summary for recipes, workflows, and custom building blocks, which includes the number of used artifacts of a selected object.

STATISTICS VIEW (SR3146.9.2.18.1)

The statistics view provides details of the following artifacts:

- Unit procedures
- Operations
- Phases
- Process parameters
- Material parameters
- MFC items (with number of inputs, outputs, transfers)
- Privilege parameters
- Capability parameters
- Work center assignment parameters
- Equipment requirement parameters
- Additional property requirements
- Transitions with identifiers
- Process parameter input expressions

STATISTICS-BASED WARNING (SR3146.9.2.18.2)

When a recipe, workflow, or custom building block with more than 1,000 phases is saved, the system displays a warning.

> A recipe, workflow, or custom building block of this magnitude exceeds the boundaries of PharmaSuite's design space and can lead to unexpected system responses.

Version Control (SR3146.4.5)

The following objects are under version control:

- master recipes,
- workflows, and
- change requests.

For the available statuses of master recipes and master workflows, see

- "Standard Statuses for Version Control (SR3146.6.4)" (page 49)

For the available statuses of change requests, see

- "Status Management of Change Requests (SR3146.12.2+)" (page 99)

Status Change (SR3146.4.5.1)

Recipe and Workflow Designer provides a status change function for master recipes, master workflows, and change requests. This includes:

- Check that master recipes and master workflows have no error or warning messages left in the **Messages (SR3146.9.2.16+)** window (page 144).
- Definition of the desired transition (target status).
- Definition of the period of validity.
  Does not apply to change requests.
- Overview of other versions of the same master recipe or master workflow, incl. their statuses and periods of validity.
  Does not apply to change requests.

Period of Validity (SR3146.6.4.2)

The period of validity of a valid object includes the **Valid from...** and **Valid until...** dates.

The start date of the validity period includes the entire day and thus begins at 12:00 AM while the end date of the validity period includes the entire day and thus lasts until 11:59 PM. This means that a validity period of exactly one day can be defined by selecting the same date for both start and end dates.

Consistency Checks (SR3146.6.4.3)

The status change checks verify the consistency of a desired status change and its related period of validity. In case of conflicts, corresponding messages are displayed.

Version History (SR3146.4.5.3)

Recipe and Workflow Designer provides a version history function for master recipes, master workflows, and change requests in order to track status transitions. For master recipes and master workflows, the version history function also tracks version transitions. The history includes transitions that took place in a different environment (e.g. prior to export or import of a master recipe or master workflow).

Configurable Version Control (SR3146.6)

The version control is configurable. The configuration includes:

- Definition of version graphs with statuses and transitions.
- Assignment of an electronic signature required to perform a status change. Electronic signatures are configurable.
- Definition of a numbering scheme for versions.

Example:

- Initial version = **V1.0**
- Status change from **Verification** to **Edit**: new version = **V1.1**
- **V1.1** is copied: new version = **V2.0**

Flexible Version Graph Assignment (SR3146.6.1)

A status graph can be assigned per class.

Example: Class Master recipe - Standard graph

Flexible Definition of Approval Details (SR3146.6.7)

The details for status changes and approvals can be defined.
For master recipes and master workflows, this includes an e-mail notification mechanism on user level.

Flexibility of User Groups (SR3146.6.7.1)

The number of user groups available in the system defines the number of different user groups that can be assigned for approval.
For further details, see use cases and RQ of electronic signatures (Security module).

Serial Approval (SR3146.6.7.3)

Serial approval of n user groups/users can be enforced.

Example 1: Group A approves, then group B can approve

For further details, see use cases and RQ of electronic signatures.

Library Concept (SR3146.9.11+)

The library concept for building blocks according to S88 includes System Building Blocks and Custom Building Blocks.

System Building Block Installation (SR3146.9.11.1)

System building block libraries can be installed.

Custom Building Block (SR3146.9.11.3)

Building blocks can be saved as custom building blocks. This applies to

- phases,
- operations,
- unit procedures, and
- procedure-level building blocks.

Chapter 10

Export/Import of Master Recipes, Master Workflows, and Custom Building Blocks (SR1075.1+)

Developing a master recipe or master workflow is an ambitious task and should be carefully planned. Thorough testing is essential before they can be approved. It is a good practice to manage their development life-cycle across multiple PharmaSuite instances (e.g. development system, test system, productive system). For this purpose, master recipes, master workflows, and custom building blocks need to be transferred from a source system to a target system.

General Prerequisites and Usage of B2MML (SR1075.1.1)

Based on B2MML, the **PharmaSuite Export/Import** functionality allows to export and imports master recipes, master workflows, and custom building blocks maintained with Recipe and Workflow Designer.
For objects that are protected by an access privilege: The system shall only allow to export objects whose access privilege matches the access privilege of the logged-in user.

The following prerequisites apply:

- Both the source and target system must be of the same (maintenance) release.
- Available system building blocks must be identical in both the source and target system.
- Any master data object that is referenced in the source system by the master recipe, master workflow, or building block to be exported must be available in the target system with the same attributes (e.g. identifier, description, version) to use the master recipe, master workflow, or building block in the target system without further manual modifications. Missing objects are detected by the **Import with missing master data (SR1075.1.3.5+)** actions (page 169).
- The versions of the underlying B2MML schemas must be identical.

Export (SR1075.1.2+)

Export Dialog (SR1075.1.2.1)

The export dialog can be opened for the master recipe, master workflow, or custom building block of the active graph window tab. This shall be possible from the **File** menu, the toolbar or the upper tab bar of the active Graph Window tab. The export functionality is protected by an export-specific access privilege.

The opened dialog shall display

- The object type of the master recipe, master workflow, or custom building block.
- The object name including the version of the master recipe or master workflow.
- The object name including the revision of the custom building block.
- The path and file name to be used for the export file containing the master recipe, master workflow, or custom building block data to be exported. Path and file name shall be changeable.
- The path and file name to be used for the log file created during the export action. Path and file name shall be changeable.
- The export result (success or error).

When the export has been performed the log file with further information shall be accessible from the export dialog.

Export File Path (SR1075.1.2.2)

The file path for the export file and the log file shall be saved user-specifically after an export has been executed and shall be provided as suggestion for the next export.

Export and Log File (SR1075.1.2.3)

The system asks for a user confirmation before overwriting an existing file.

Export File Data (SR1075.1.2.4)

The export file shall contain all data to be able to create the exported master recipe, master workflow, or custom building block in a target system if the target system fulfills the prerequisites stated by the **General Prerequisites and Usage of B2MML (SR1075.1.1)** requirement (page 165).

Export File Checksum (SR1075.1.2.4.1)

The system shall add a checksum to the export file to allow the import functionality to detect a manipulation of the file content.

VERSION AND STATUS OF EXPORTED DATA (SR1075.1.2.4.2)

The version and status of the exported objects in the source system are part of the export file data.

STATUS HISTORY OF EXPORTED DATA (SR1075.1.2.4.3)

The export file data for the master recipe, master workflow, or custom building block shall contain the information about the source system that holds the history data of the latest status transition prior to the export.

Import (SR1075.1.3+)

Import Dialog (SR1075.1.3.1)

The import dialog for a master recipe, master workflow, or custom building can be opened from the **File** menu or the toolbar. Both actions are protected by an import-specific access privilege.

The opened dialog shall provide

- A file selector to select the file with the data to be imported.
- A file selector to define the path and file name for the log file.

After selecting an import file, the dialog shall display

- The object type of the object to be imported.
- The object name to be imported including the version of the master recipe or master workflow.
- The object name to be imported including the revision of the custom building block.
- The import result (success, with warnings, or error)

When the import has been performed the log file with further information shall be accessible from the import dialog.

Import File Path (SR1075.1.3.2)

The file path for the import file and the log file shall be saved after an import action and provided as suggestion in the file selector for the next import.

Import Log File (SR1075.1.3.3)

The system asks for a user confirmation before overwriting an existing log file.

Import of Data (SR1075.1.3.4)

Independent of the current active workbench, it shall be possible to import a master recipe, a master workflow, or a custom building block.

Only if the imported object matches the currently active workbench and the import has not failed, is the object opened in a new Graph Window tab. If the object is protected by an access privilege and this access privilege does not match the access privilege of the logged-in user, the opening of the new tab is omitted.

If the import file contains choice elements that are not available in the target system, the import fails and the missing choice elements are listed in the log file.

Import File Checksum(SR1075.1.3.4.1)

The system shall check whether the export checksum still matches the file content. If not, the import fails.

Version and Status of Imported Object (SR1075.1.3.4.2)

The master recipe, master workflow, or custom building block is imported with the version and status as exported. If the object already exists in the target system, the import fails.

A status transition can be skipped, see the **Skip status transition (SR1075.1.3.5.4)** action (page 170).

Status History of Imported Data (SR1075.1.3.4.3)

The master recipe, master workflow, or custom building block is imported with the version and status as exported. An exception to this can be caused by the **Skip status transition action (SR1075.1.3.5.4)** action (page 170). If the object already exists in the target system, the import fails.

Import with Missing Master Data (SR1075.1.3.5+)

A master recipe, master workflow, or custom building block can be imported even if referenced master data is not available in the target system. To support this, the system executes several actions:

- **Replace a Non-existing Object with a Dummy Object (SR1075.1.3.5.1)** action (page 169)
- **Remove an Object Link to a Non-existing Object (SR1075.1.3.5.2)** action (page 169)
- **Detect and Remove a String Reference to a Non-existing Object (SR1075.1.3.5.3)** action (page 170)
- **Skip Status Transition (SR1075.1.3.5.4)** action (page 170)

If these actions have adapted the master recipe, master workflow, or custom building block and have removed references to allow it to be imported, all adaptions are documented with a warning:

- **Import Consistency Check Messages (SR1075.1.3.5.5+)** warnings (page 170)

Replace a Non-existing Object with a Dummy Object (SR1075.1.3.5.1)

The Material (product) of a master recipe must be available in the target system. It can be an object reference to an ERP BOM or to a material. If the material or the ERP BOM is not available in the target system, it is replaced with a dummy material created by PharmaSuite (“S88DefImpMaterial”).

Remove an Object Link to a Non-existing Object (SR1075.1.3.5.2)

A master recipe, master workflow, or custom building block to be imported can contain object links to master data. If the linked master data is not available at the target system, the object link and all non-allowed data structures are removed.

Detect and Remove a String Reference to a Non-existing Object (SR1075.1.3.5.3)

A master recipe, master workflow, or custom building block can contain soft references to master data defined with phase parameters. Soft references to non-available objects are not detected by the common Recipe and Workflow Designer consistency checks and will cause unexpected system behavior during execution.
The system shall execute checks at import to detect as many of these soft references to non-available objects as possible.
Since the PharmaSuite system integrators can build their own system phases, the system shall be extendable by a PharmaSuite system integrator to provide and configure additional checks to detect a soft reference to a non-available object (see chapter "Implementing Checks of Soft References" in the "Technical Manual Developing System Building Blocks" [A6] (page 171)).
If a check detects a soft reference to a non-available object in the target system, the soft reference is deleted.

Skip Status Transition (SR1075.1.3.5.4)

An imported master recipe, master workflow, or custom building block is kept in the **Draft** status in case it was changed to allow to be imported even if not all required master data objects were available in the target system.

Import Consistency Check Messages (SR1075.1.3.5.5+)

The following warning messages are available to inform the recipe or workflow author about issues found and actions taken by the import consistency checks. They are displayed in the **Messages (SR3146.9.2.16+)** window (page 144).

Import check – Missing data in target system (SR1075.1.3.5.5.1)

Message category: Warning

Issue: The imported master recipe, master workflow, or custom building block references data that is not available in the target system.

The warning can be removed with the **Messages - Resolve issue (SR3146.9.2.16.4)** function (page 146). The system does not create missing data, it only removes the warning.

Abort Import if a Confidential Object Cannot Be Protected (SR1075.1.3.6)

A master recipe, master workflow, or custom building block can be protected by an access privilege. If the access privilege is not available in the target system, the import shall be aborted to avoid unprotected access to the confidential object.

Chapter 11

Reference Documents

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

No.	Document Title	Part Number
A1	PharmaSuite Functional Requirement Specification Data Management	PSFRSDM-RM006C-EN-E
A2	PharmaSuite Functional Requirement Specification Execution Framework	PSFRSEF-RM006C-EN-E
A3	PharmaSuite Functional Requirement Specification Runtime Data Management	PSFRSRT-RM006C-EN-E
A4	PharmaSuite Functional Requirement Specification Non-functional Requirements	PSFRSNF-RM003C-EN-E
A5	PharmaSuite Technical Manual Configuration & Extension - Volume 3	PSCEV3-GR010C-EN-E
A6	PharmaSuite Technical Manual Developing System Building Blocks	PSBB-PM010B-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
Norbert Ern	Product Owner
Wolfgang Schmitt	Technical Lead
Ignaz Wangler	Test Lead

Version Information

Object	Version
PharmaSuite	10.02.00
Functional Requirement Specification	1.0

Revision History

The following tables describe the history of this document.

Changes related to the document:

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

Changes related to "Recipe and Workflow Management" (page 3):

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

Changes related to "Management of Master Recipes and Master Workflows" (page 7):

Object	Description	Document
Process Inputs - Escalation-enabled Capability (SR3146.9.5.9.3) (page 24)	Update System actions added: Run created, Finish due reminder, Finish overdue. Action attributes added: Alarm enabled, Notification text. Context information-related It is not expected that HTML tags are used as part of the notification or exception text. This means that the notification text will be shown as plain text and HTML formatting will be not interpreted by the notification panel.	1.0
Locked Transitions (SR3146.9.14.1) (page 32)	Update The identifier of a transition is always read-only regardless of whether the transition is locked or not, if the transition is part of a used custom building block in a read-only status.	1.0
Validator - Simultaneous Branch – End Twice (SR3146.9.4.3.148) (page 56)	New section	1.0

Changes related to "Management of Building Blocks" (page 83):

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

Changes related to "Comparing Master Recipes, Master Workflows, and Custom Building Blocks" (page 93):

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

Changes related to "Management of Change Requests" (page 97):

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

Changes related to "Master Recipe Report - Batch" (page 115):

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

Changes related to "Master Workflow Report" (page 123):

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

Changes related to "Designer Workbench" (page 129):

Object	Description	Document
Messages (SR3146.9.2.16+) (page 144)	Context information-related They result from checks performed by the Designer workbench when master recipes, master workflows, or building blocks are being edited or imported.	1.0
Messages - Resolve Issue (SR3146.9.2.16.4) (page 146)	Update For messages caused by inconsistencies that can be resolved automatically, the system provides the Resolve issue functions in the context menu of the related message to resolve the selected issue, to resolve all issues of the same message category identifier, or to resolve all issues. Context information-related The **Resolve issue** function is available for the following checks and warnings: **Import Consistency Check Messages (SR1075.1.3.5.5+)** warnings	1.0
Unit Procedure and Operation Path Editor (SR3146.9.2.17.5) (page 160)	New section	1.0

Changes related to "Export/Import of Master Recipes, Master Workflows, and Building Blocks" (page 165):

Object	Description	Document
General Prerequisites and Usage of B2MML (SR1075.1.1) (page 165)	Update **PharmaSuite Import/Export** functionality of Recipe and Workflow Designer Any master data object that is referenced in the source system by the master recipe, master workflow, or building block to be exported must be available in the target system with the same attributes (e.g. identifier, description, version) to use the master recipe, master workflow, or building block in the target system without further manual modifications. Missing objects are detected by the **Import with missing master data (SR1075.1.3.5+)** actions.	1.0
Prerequisites For Import (SR1075.1.1.1)	Deleted	1.0
Checksum (SR1075.1.1.2)	Deleted	1.0
Handling of Status and Version (SR1075.1.1.3)	Deleted	1.0
Handling of Status Transition History (SR1075.1.1.4)	Deleted	1.0
Export (SR1075.1.2+) (page 166)	New section , Export File Path (SR1075.1.2.2), Export and Log File (SR1075.1.2.3), Export File Data (SR1075.1.2.4), Export File Checksum (SR1075.1.2.4.1), Version and Status of Exported Data (SR1075.1.2.4.2), Status History of Exported Data (SR1075.1.2.4.3)	1.0
Export Dialog (SR1075.1.2.1) (page 166)	New section	1.0
Export File Path (SR1075.1.2.2) (page 166)	New section	1.0
Export and Log File (SR1075.1.2.3) (page 166)	New section	1.0
Export File Data (SR1075.1.2.4) (page 166)	New section	1.0
Export File Checksum (SR1075.1.2.4.1) (page 166)	New section	1.0
Version and Status of Exported Data (SR1075.1.2.4.2) (page 167)	New section	1.0

Object	Description	Document
Status History of Exported Data (SR1075.1.2.4.3) (page 167)	New section	1.0
Import (SR1075.1.3+) (page 167)	New section	1.0
Import Dialog (SR1075.1.3.1) (page 167)	New section	1.0
Import File Path (SR1075.1.3.2) (page 167)	New section	1.0
Import Log File (SR1075.1.3.3) (page 167)	New section	1.0
Import of Data (SR1075.1.3.4) (page 168)	New section	1.0
Import File Checksum(SR1075.1.3.4.1) (page 168)	New section	1.0
Version and Status of Imported Object (SR1075.1.3.4.2) (page 168)	New section	1.0
Status History of Imported Data (SR1075.1.3.4.3) (page 168)	New section	1.0
Import with Missing Master Data (SR1075.1.3.5+) (page 169)	New section	1.0
Replace a Non-existing Object with a Dummy Object (SR1075.1.3.5.1) (page 169)	New section	1.0
Remove an Object Link to a Non-existing Object (SR1075.1.3.5.2) (page 169)	New section	1.0
Detect and Remove a String Reference to a Non-existing Object (SR1075.1.3.5.3) (page 170)	New section	1.0
Skip Status Transition (SR1075.1.3.5.4) (page 170)	New section	1.0
Import Consistency Check Messages (SR1075.1.3.5.5+) (page 170)	New section	1.0

Object	Description	Document
Import Check – Missing Data in Target System (SR1075.1.3.5.5.1) (page 170)	New section	1.0
Abort Import if a Confidential Object Cannot Be Protected (SR1075.1.3.6) (page 170)	New section	1.0

Index

P

R

S

V

W