LISTEN.
THINK.
SOLVE.®

PharmaSuite®

RECIPE AND WORKFLOW DESIGNER
VOLUME 3: RECIPE DESIGNER - DEVICE
RELEASE 8.4
USER MANUAL

AB Allen-Bradley · Rockwell Software

Rockwell
Automation

Contact Rockwell See contact information provided in your maintenance contract.

Contents

Recipe Designer - Device

The Recipe Designer - Device application of Recipe and Workflow Designer provides an environment for building and maintaining master recipes, their component building blocks, and change requests for mass changes of master recipes.
Its basic screen layout consists of the following areas:

- menus (page 17) and toolbars (page 26) for accessing the functions of Recipe Designer - Device

- Graph Window (page 29) with upper and lower tab bar navigation

- two navigation support windows, Map (page 33) and Explorer (page 34)

- Setlist (page 35) window for graph-building support

- Property (page 45) windows for displaying the properties of recipe and building block elements, their headers, and source building blocks

- Phase Preview (page 55) window for displaying how a phase will be rendered during execution

- Messages (page 57) window for displaying all error, warning, or information messages pertaining to the active master recipe or building block

- Comparison (page 67) window displaying all differences between the active master recipe or building block and its baseline

- status bar (page 82) with general information.
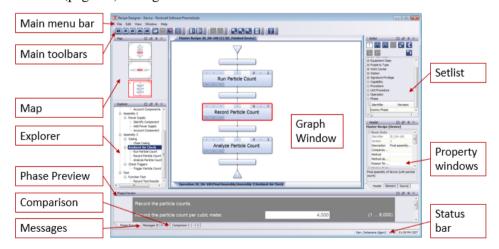
Figure 1: Screen layout of Recipe Designer - Device

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).

Italic typeface   Designates technical background information, such as

- path, folder, and file names

- methods

- classes.

CAPITALS   Designate keyboard-related information, such as

- key names

- keyboard shortcuts.

`Monospaced typeface`   Designates code examples.

TIP

Instructions in this manual are based on Windows 7. Select the appropriate commands if you are using a different operating system.

Basic Operations

The following sections describe basic and recurring operations and functions in Workflow Designer.

Data Tools

When you design recipes, workflows, or building blocks, you are working on individual data records that are stored in the database, which you need to access. For these situations, Recipe and Workflow Designer supports you with tools for searching (page 13) and filtering (page 14) lists of data objects it has retrieved from the database and provides sorting (page 14) functions you can apply to all tabular displays of data objects.
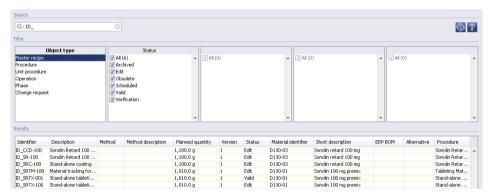
Figure 2: Search and Filter tools with sortable Results list

Searching

The **Search** tool provides you with a fast and efficient way to locate items with specific content in the pre-filtered **Results** list:

- In the input box, type the string of characters that will be applied to the **Results** list. The search becomes effective with the second character.

- To clear the input box, click the **Cancel** icon that appears when you have typed the first character in the box.

TIP

By default, the search is not case-sensitive and runs over all columns of the **Results** list. Click the magnifying glass icon to restrict the search to specific columns, set the search criteria to be case-sensitive, or allow the use of wildcards.

Filtering

The **Filter** tool is hierarchically organized into search categories that make refining your search fast and easy.

1. In the first column, select the type of the database objects you are searching. The system displays all database objects of this type in the **Results** list.

> **TIP**
>
> If the **Object type** is defined by the context from which you have opened the form, the column only provides corresponding values.

2. Depending on the selected object type, the system provides further search categories.
   Select or unselect one, several, or all of the options available in the first column to the right of the **Object type** to reduce the number of objects displayed in the **Results** list.

3. To further refine your search result, move to the next category column and select or unselect its options as required. The system applies your choices as additional filter and displays the subset of remaining objects in the **Results** list.

> **TIP**
>
> Please note that Recipe Designer takes a snapshot of the database when you access an object type. This means that the filter tool will remain unaware of changes made to database objects or new objects added by other users while you browse through the data of this object type. To synchronize the snapshot with the central database, click the **Refresh from database** button.

Sorting

By default, the sort order of a data table depends on the type of data objects it displays. It is, however, possible to change the sort order, sort by another column, and sort by two or more columns as primary, secondary, and further levels. Additionally you can reorder the table columns themselves.

■ To adjust the sorting, proceed as follows:

1. Click any column header to sort the table by this column in ascending order. The system indicates the sort order with a triangle pointing up.

2. Re-click the same column header to switch the sort order to descending. The system indicates the sort order with a triangle pointing down.

3. CTRL-click a yet unmarked column header to add this column as further sort level in ascending sort order, indicated by the triangle pointing up and the count number indicating the sort level.

4. Re-CTRL-click the same column header to switch the sort order to descending without changing its sort level.

■ To reorder the table columns, click a column header and drag it to the desired position in the table.

Signature Requests

When performing safety-sensitive or GxP-relevant functions the system may request you to enter an electronic signature, for example during a status change. Signatures are linked to user groups and access privileges, which means that the system will only accept the signature of a user who is qualified to perform the task in question. Unless the required signature data has been entered correctly, subsequent functions cannot be executed.

For situations requiring a witness, the system will ask not only for a single but for a double signature. In these cases two different users, typically with different qualifications, have to complete the signature form before task processing can continue.

To perform an electronic signature, type your login name and password and click the **OK** button. Comments can be optional or mandatory and may consist of up to 255 characters.
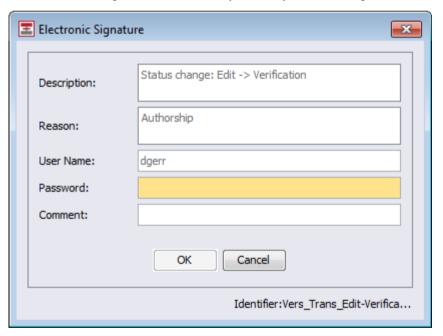
Figure 3: Single electronic signature

Figure 4: Double electronic signature to support witness role

Menus and Toolbars

You can access all relevant functions either from the menu bar in the main Recipe Designer window, from context-sensitive shortcut menus, or from toolbars provided for quick access to frequently used functions or for graph building.

Menus

Recipe Designer provides a main menu bar (page 17) with all relevant functions as well as context-sensitive shortcut menus (page 24) for quick access to specific functions.

MAIN MENU BAR

The **main menu bar** offers the following menus and functions:

File

■ New master recipe (CTRL+SHIFT+M)
Creates a new master recipe, thus first opening the **Select material** dialog (page 84), then the **New Master Recipe** dialog to define the recipe's identifier, and afterwards a new blue tab in the upper tab bar.

■ New procedure (CTRL+SHIFT+R)
Creates a new procedure, thus first opening the **New Procedure** dialog (see "New Building Block" in Vol. 1) and afterwards a new yellow tab in the upper tab bar.

■ New unit procedure (CTRL+SHIFT+U)
Creates a new unit procedure, thus first opening the **New Unit Procedure** dialog (see "New Building Block" in Vol. 1) and afterwards a new yellow tab in the upper tab bar.

■ New operation (CTRL+SHIFT+O)
Creates a new operation, thus first opening the **New Operation** dialog (see "New Building Block" in Vol. 1) and afterwards a new yellow tab in the upper tab bar.

■ New phase (CTRL+SHIFT+F)
Creates a new phase, thus first opening the **New Phase** dialog (see "New Building Block" in Vol. 1) and afterwards a new yellow tab in the upper tab bar.

■ New change request (CTRL+SHIFT+C)
Creates a new change request, thus first opening the **New Change Request** dialog and afterwards a new orange tab in the upper tab bar.

- Open (CTRL+O)
  Opens the **Open** dialog (page 83) to select a component to be loaded into the
  Graph Window (page 29).

- Save <component identifier> (CTRL+S)
  Saves all changes made to the main component that is currently active in the
  upper tab bar. This action also saves all of its subordinate components that may be
  open in the lower tab bar.

- Save <component identifier> as (CTRL+F12)
  Opens the **Save <Component Type> as** dialog to save the main component that
  is currently active in the upper tab bar under a new identifier. Along with saving
  the main component, this action also saves all of its subordinate components that
  may be open in the lower tab bar.

- Save all (CTRL+SHIFT+S)
  Saves all changes made to all main components that are currently open in the
  upper tab bar. This action also saves all of their subordinate components that may
  be open in the lower tab bar.

- Close <component identifier> (CTRL+F4)
  Closes the main component tab that is currently active in the upper tab bar. This
  action also closes all of its subordinate components tabs that may be open in the
  lower tab bar.

- Close all (CTRL+SHIFT+F4)
  Closes all main components tabs that are currently open in the upper tab bar. This
  action also closes all of their subordinate components tabs that may be open in the
  lower tab bar.

- Delete <component identifier>
  Deletes the main component that is currently active in the upper tab bar.

- Rename <component identifier>
  Opens the **Rename <Component Type>** dialog where you can type a new
  identifier.

- Replace material (product)
  Opens the **Select material** dialog (page 84) where you can select another material
  or material with ERP BOM as product of your master recipe.

- Prepare <component identifier> for status change
  Only available for saved master recipes, change requests, or building blocks,
  performs an internal preparation on a complete component that has passed all
  checks. The preparation is necessary to make the component available for status
  changes.

- Change status of <component identifier> (CTRL+H)

  - For master recipes, opens the **Change Status** dialog to perform a status change (page 88) on the master recipe that is currently active in the upper tab bar.

  - For change requests, opens the **Change Status** dialog to perform a status change (page 92) on the change request that is currently active in the upper tab bar.

  - For building blocks, opens an **Electronic Signature** dialog to perform and sign a status change (see "Changing the Status of Building Blocks" in Vol. 1) on the building block that is currently active in the upper tab bar.

- View status history of <component identifier> (CTRL+Q)
  Opens the **Status History** window to display a list of all status changes that have been performed on the component that is currently active in the upper tab bar. Only available for master recipes (page 90) and change requests (page 93).

- Print report (CTRL+P)
  Opens the **Master Recipe Report** of the master recipe that is currently active in the upper tab bar.
  Only available for saved master recipes.

- Print report (without comparison result) (CTRL+SHIFT+P)
  Opens the **Master Recipe Report** of the master recipe that is currently active in the upper tab bar. The report does not include the **Comparison with Baseline** section.
  Only available for saved master recipes.

- Open graph pagination (CTRL+R)
  Opens the **Graph Pagination** dialog to display the tiled print preview for the SFC graph that is currently active in the Graph Window.
  Only available for master recipes.

- Compile usage list of <building block identifier>
  Opens the **Usage List** dialog to search and list all occurrences of the building block that is currently active in the upper tab bar.
  Only available for building blocks.

- Show statistics of <component identifier>
  Opens the **Statistics of <Component Identifier>** dialog to display a list that shows how many structure elements the component holds that is currently active in the upper tab bar.

- Exit (ALT+F4)
  Closes the application window.

> **TIP**
>
> To create a new version of a master recipe, use the **Save as** function to save it under the same identifier. The system will ask you if you wish to create a new version. Confirm with **Yes** to save the master recipe with the same identifier and the next-higher version number.
>
> To create a new revision of a building block, use the **Save as** function to save it under the same identifier but type a new value in the **Revision** box.
>
> The version or revision numbers of components are shown in square brackets in their tab titles in the upper tab bar of the Graph Window (page 29).

Edit

- Rename selected (F2)

  In an SFC graph tab, makes the identifier of the currently selected component available for editing. Press the ESC key to cancel this action.

- Select all (CTRL+A)

  Selects all components of the currently active SFC or MFC graph.

- Clear selection (CTRL+D)

  Unselects all current selections made in the currently active SFC or MFC graph.

- Select steps and transitions (CTRL+T)

  In an SFC graph tab, selects all steps and transitions of the currently active graph.

- Select links (CTRL+L)

  In an SFC graph tab, selects all links of the currently active graph.

- Unselect steps and transitions (CTRL+SHIFT+T)

  In an SFC graph tab, unselects all steps and transitions of the currently active graph.

- Unselect links (CTRL+SHIFT+L)

  In an SFC graph tab, unselects all links of the currently active graph.

- Invert selection (CTRL+I)

  Inverts all selections made in the currently active SFC or MFC graph. This action unselects all selected components and selects all unselected components.

- Undo (CTRL+Z)

  In an SFC graph tab, revokes the last action you have performed. You can undo up to 100 actions, thus you can step by step revoke the last 100 action you have performed.

  > **TIP**
  > Please note that the stepwise **Undo** function does not work on changes you have performed on parameters through the Parameter Panel.

■ Redo (CTRL+Y)
In an SFC graph tab, redoes the last action revoked with the **Undo** action.
You can redo up to 100 actions, thus you can step by step redo the last 100 actions you have revoked.

■ Delete selected (DEL)
In an SFC graph tab, deletes all currently selected components.

■ Find (CTRL+F)
In an SFC graph tab, opens the **Find Component** dialog to type a sequence of characters to search for in the components of the currently open graph.

■ Find again (F3)
In an SFC graph tab, finds the next occurrence in a component of the characters you typed in the **Find Component** dialog box.

View

■ Zoom to fit (CTRL+PERIOD)
Sets the zoom factor of the currently active graph to fit the graph on the available screen space of the Graph Window.

■ Zoom to 100% (CTRL+1)
Resets the zoom factor of the currently active graph to its default value.

■ Zoom in (CTRL+PLUS)
Zooms in on the graph in the currently active tab, doubling its display size.

■ Zoom out (CTRL+MINUS)
Zooms out from the graph in the currently active tab, reducing its size to half of its previous display size.

■ Show/hide grid (CTRL+G)
Toggles the display of grid lines in the Graph Window.

■ Map (ALT+M)
Toggles the display of the Map (page 33) for the currently active component.

■ Explorer (ALT+E)
Toggles the display of the Explorer (page 34) for the currently active component.

■ Setlist (ALT+S)
Toggles the display of the Setlist (page 35) for your current session.

■ Element properties (ALT+R)
Toggles the display of the **Element** property window (page 52) of the currently active component.

■ Source properties (ALT+B)
Toggles the display of the **Source** property window (page 53) of the currently active component.

- ■ Header properties (ALT+H)
  Toggles the display of the **Header** property window (page 45) of the currently active component.

- ■ Phase Preview (ALT+P)
  Toggles the display of the Phase Preview (page 55) for the currently active component.

- ■ Messages (ALT+V)
  Toggles the display of the Messages (page 57) window for the currently active component.

- ■ Comparison (ALT+O)
  Toggles the display of the Comparison (page 67) window for the currently active component.

- ■ Universe (ALT+U)
  Opens the Universe (page 94) for your current session to select a component for loading it into the Setlist.

- ■ Material Flow Control (ALT+C)
  Opens the **Material Flow Control** tab (page 158) for the currently active master recipe or procedure building block.

Window

- ■ Recipe Designer - Batch
  Switches to the Recipe Designer - Batch mode of Recipe and Workflow Designer. Before switching, the system closes all components currently open in the Graph Window, prompting you to save if you have unsaved changes.
  A checkmark to the left of the Recipe Designer - Batch, Recipe Designer - Device, or Workflow Designer menu options indicates which mode is currently active.
  Only available if you have the user rights necessary for accessing Recipe Designer - Batch.

- ■ Recipe Designer - Device
  Switches to the Recipe Designer - Device mode of Recipe and Workflow Designer. Before switching, the system closes all components currently open in the Graph Window, prompting you to save if you have unsaved changes.
  A checkmark to the left of the Recipe Designer - Device, Recipe Designer - Batch, or Workflow Designer menu options indicates which mode is currently active.
  Only available if you have the user rights necessary for accessing Recipe Designer - Device.

■ Workflow Designer
Switches to the Workflow Designer mode of Recipe and Workflow Designer. Before switching, the system closes all components currently open in the Graph Window, prompting you to save if you have unsaved changes.
A checkmark to the left of the Workflow Designer, Recipe Designer - Batch, or Recipe Designer - Device menu options indicates which mode is currently active. Only available if you have the user rights necessary for accessing Workflow Designer.

> **TIP**
> Please note that the system closes all open objects when you switch between modes. So after a mode switch, the Graph Window is empty as if you had started the application itself.

■ Undo layout change
Revokes the last layout change you have performed. You can undo up to 100 actions, thus you can step by step revoke the last 100 layout changes you have performed. Once you have performed a layout change, the menu function changes to a more precise description of the undo action, such as **Undo resizing** or **Undo dragging**.

■ Redo layout change
Redoes the last layout change you have revoked with the **Undo layout change** function. You can redo up to 100 actions, thus you can step by step redo the last 100 layout changes you have revoked. Once you have performed a layout change, the menu function changes to a more precise description of the undo action, such as **Redo resizing** or **Redo dragging**.

■ Save user layout
Saves the current window layout and overwrites the layout that was last saved by you on this computer.

■ Load user layout
Loads the last layout you have saved on this computer with the **Save user layout** function.

■ Reset layout
Resets the window layout to the system-defined default layout. This function does not affect the saved user layout, which can be restored with the **Load user layout** function.

Help

■ Recipe Designer Help (ALT+F1)
Opens the start page of the help system (page 127).

■ About PharmaSuite
Opens the **About PharmaSuite** dialog (page 129).

SHORTCUT MENUS

There are four **shortcut menus** available, which you can access by right-clicking in the following screen locations:

MFC graph tab of the Graph Window

■ Close sub-tab (CTRL+W)
Closes the currently active tab of the lower tab bar.

> **TIP**
> Please note that the functions for merging or splitting are not available for device recipes.

SFC graph tab of the Graph Window

■ New step (unconnected) (INS)
Inserts a new unconnected step at the current cursor position.

■ Rename selected (F2)
Makes the identifier of the currently selected component available for editing. Press the ESC key to cancel this action.

■ Create building block from selected (F12)
Opens the **Create <Object Type>** dialog (see "Create Building Block" in Vol 1) where you can save your parameterized building block to make it available for selection from the Universe.

■ Delete selected (DEL)
Deletes all currently selected components.

■ Arrange
Arranges the graph and its components to center sequences and branches with respect to their start components and to optimize the vertical spacing between individual components.

■ Compact graph
Compacts the graph for optimal use of space.

■ Go back (ALT+LEFT)
Returns to the previously active tab in the lower tab bar within the currently active main component.

■ Go up (ALT+UP)
Moves to the next higher structure level.

■ Close sub-tab (CTRL+W)
Closes the currently active tab of the lower tab bar.

Upper tab bar of the Graph Window

- New master recipe (CTRL+SHIFT+M)
  Creates a new master recipe, thus first opening the **Select material** dialog (page 84), then the **New Master Recipe** dialog to define the recipe's identifier, and afterwards a new blue tab in the upper tab bar.

- New procedure (CTRL+SHIFT+R)
  Creates a new procedure, thus first opening the **New Procedure** dialog (see "New Building Block" in Vol. 1) and afterwards a new yellow tab in the upper tab bar.

- New unit procedure (CTRL+SHIFT+U)
  Creates a new unit procedure, thus first opening the **New Unit Procedure** dialog (see "New Building Block" in Vol. 1) and afterwards a new yellow tab in the upper tab bar.

- New operation (CTRL+SHIFT+O)
  Creates a new operation, thus first opening the **New Operation** dialog (see "New Building Block" in Vol. 1) and afterwards a new yellow tab in the upper tab bar.

- New phase (CTRL+SHIFT+F)
  Creates a new phase, thus first opening the **New Phase** dialog (see "New Building Block" in Vol. 1) and afterwards a new yellow tab in the upper tab bar.

- Open (CTRL+O)
  Opens the **Open** dialog (page 83) to select a component to be loaded into the Graph Window (page 29).

- Close <component identifier> (CTRL+F4)
  Closes the main component tab that is currently active in the upper tab bar. This action also closes all of its subordinate components tabs that may be open in the lower tab bar.

- Close all but <component identifier> (CTRL+ALT+F4)
  Closes all main component tabs except for the main component tab that is currently active in the upper tab bar.

- Save <component identifier> (CTRL+S)
  Saves all changes made to the main component that is currently active in the upper tab bar. This action also saves all of its subordinate components that may be open in the lower tab bar.

- Rename <component identifier>
  Opens the **Rename <Component Type>** dialog where you can type a new identifier.

- Delete <component>
  Deletes the main component that is currently active in the upper tab bar.

Message in the Messages Window for which a direct resolution is available

- Resolve issue
  Directly resolves the issue indicated in the message, thus removing the message from the list.

- Resolve all issues (<number of issues>)
  Resolves all issues listed in the panel for which a direct resolution is available. In parenthesis, the menu function displays the number of issues it is going to resolve.

Toolbars

Recipe Designer provides several toolbars for context-sensitive quick access to specific functions.

Main Toolbars

The **main toolbars** (see "Toolbar Buttons" in Vol. 1) offer shortcuts to the more frequently required menu functions:

- File toolbar with
  - New master recipe
  - New procedure
  - New unit procedure
  - New operation
  - New phase
  - Open
  - Save
  - Close
  - Open graph pagination
- Status toolbar with
  - Change status
  - View status history
- Edit toolbar with
  - Undo
  - Redo

- View toolbar with
  - Zoom to 100%
  - Zoom in
  - Zoom out
  - Open Universe
- Help toolbar with
  - Help

Process Workflow Toolbar

The **process workflow toolbar** (see "Toolbar Buttons" in Vol. 1) opens when you right-click an SFC graph element in the work area and provides the following functions for building SFC graph structures:

- Sequence
- Selection branch
- Simultaneous branch
- Join
- Loop

Setlist Toolbar

The **Setlist toolbar** (see "Toolbar Buttons" in Vol. 1) provides the following functions for building recipes or building blocks with pre-selected building blocks from the Setlist:

- Sequence
- Selection branch
- Simultaneous branch
- Join
- Replace
- Smart replace
- Input
- Output
- Invert drawing direction

Rockwell Software PharmaSuite® - Recipe Designer - Device

Graph Window

The Graph Window is the main work area of Recipe Designer. It is where you build recipes (page 131) with all their component graphs on the different levels of the recipe structure, create and manage custom building blocks (see "Managing Building Blocks" in Vol. 1), and define and process change requests.

When you open a recipe (page 131) or create a new recipe (page 131), the system opens a new tab in the upper tab bar above the Graph Window. The tab title has a blue background and displays the following information:

- the object type (master recipe)

- the recipe's identifier

- the recipe's version number, which displays as 0 until a new recipe has been saved for the first time

- the identifier of the material it produces

- a status marker, if applicable.
  An asterisk (*) indicates that the recipe has unsaved changes, while recipes that are not editable are marked by an (R).

On the main tab of the master recipe, all of its graphs, such as the recipe graph with its single procedure element, the various graphs on subordinate structure levels, or its MFC graph and data, are displayed as individual sub-tabs, which are located on the lower tab bar under the Graph Window.

> **TIP**
>
> Whether or not you can edit a master recipe depends on its status. Only master recipes in the **Edit** status can be edited. The system indicates that a recipe is not editable by displaying all of its Graph Window tabs with a gray background color.

When you open a building block or create a new building block (see "Create Building Block" in Vol 1), the system opens a new tab in the upper tab bar above the Graph Window. The tab title has a yellow background and displays the following information:

- the object type (procedure, unit procedure, operation, phase)

- the building block's identifier

■ a status marker, if applicable.
An asterisk (*) indicates that the building block has unsaved changes, while building blocks that are not editable are marked by an (R).

On the main tab of the building block, all of its graphs, such as the building block's component view with its single element, or the various graphs on subordinate structure levels, are displayed as individual sub-tabs, which are located on the lower tab bar under the Graph Window.

> **TIP**
>
> **Approved** or **Archived** building blocks are not editable. This is not only indicated by the (R) marker but also by their gray background color in the Graph Window.

When you open a change request or create a new change request, the system opens a new tab in the upper tab bar above the Graph Window. The tab title has an orange background and displays the following information:

■ the object type (change request)

■ the change request's identifier

■ a status marker, if applicable.
An asterisk (*) indicates that the change request has unsaved changes, while change requests that are not editable are marked by an (R).

You can have several recipes, building blocks, and/or change requests open simultaneously and switch between them by clicking their respective tabs in the upper tab bar.

When the number of open tabs on a tab bar exceeds the available width of the tab bar, the system displays the following buttons to support your navigation:

◀ Scroll backward
Scrolls the tab bar to the left.

▶ Scroll forward
Scrolls the tab bar to the right.

▤ Close
Displays a list of all open tabs from which you can select a tab you wish to close. The currently active tab is indicated by bold font.

> **TIP**
>
> Building block title tabs are yellow and their work area has a yellow background to differentiate them from recipe tabs that have blue title tabs and a plain white background in the work area.

Panning and Zooming the Graph Window

You can navigate any graph view by panning and zooming:

- To pan or move your current graph view, click anywhere on the background of the Graph Window and drag the graph into the desired direction.

- To zoom in (enlarge) on your graph view, centered on your current mouse cursor position, scroll your mouse wheel forward.

- To zoom in (enlarge) on your graph view, centered on the Graph Window, CTRL+scroll your mouse wheel forward.

- To zoom out from your graph view, centered on your current mouse cursor position, scroll your mouse wheel backward.

- To zoom out from your graph view, centered on the Graph Window, CTRL+scroll your mouse wheel backward.

Selecting Components in the Graph Window

To select one or more components in your graph view you can either select each component individually or marquee-select a group of components by drawing a frame around them.

To select individual components, proceed as follows:

- To select a single component, left-click it.

- To add a single component to an existing selection, SHIFT+left-click it.

- To toggle the selection of a single component on or off, CTRL+left-click it.

To marquee-select a group of components, proceed as follows:

- To draw a marquee around a group of components, SHIFT+left-click on the background of the Graph Window and drag the frame until it fully covers all components you wish to select.

> **TIP**
> Please note that the marquee always has a rectangular shape. There is no freehand marquee available.

- To toggle the selection of a group of components, CTRL+left-click on the background of the Graph Window and drag the frame until it fully covers all selected components you wish to unselect.

- To extend an existing selection, CTRL+left-click on the background of the Graph Window and drag the frame until it fully covers all unselected components you wish to add to your current selection.

Navigation Support Windows

Recipe Designer provides two dockable windows to support your navigation through your recipes, building blocks, and their components.

Map

The Map window provides an overview of the graph structure you are working on. It consists of three overview frames, one for each structure level that holds a graph. The structure levels are differentiated by the number of corners that have a line drawn across them:

■ the procedure level with the graph of unit procedures, indicated by four crossed corners

■ the unit procedure level with the graph of operations, indicated by three crossed corners

■ the operation level with the graph of phases, indicated by two crossed corners.

Each frame shows the entire graph on its respective structure level, depending on the hierarchy level of the currently active tab in the Graph Window. So, if you have a procedure tab of a master recipe active in your Graph Window, you only see the graph of unit procedures in the top frame, while the two lower frames remain empty. On the other hand, if you have an operation tab active in your Graph Window, you see the graph of phases in the bottom frame, the graph of operations in the middle frame, and the graph of unit procedures in the top frame.
If you open a building block, the Map first displays the building block's header on the respective structure level. All higher levels naturally remain empty.
The frame that displays the active Graph Window tab is indicated by a colored border. In the frames that show a higher level of the hierarchy, a colored border indicates the parent component of the respective lower graph.

A Map frame displays a selection frame that reflects the graph section visible in the Graph Window. If the graph section is large enough, the selection frame only becomes visible when you change the zoom factor or pan the graph section. To use the frames of the Map for panning, zooming, and navigating the graphs of your recipe, proceed as follows:

■ SHIFT+click and drag to draw a new selection frame whose content will be displayed in the Graph Window.

- Click and drag to move an existing selection frame, thus panning the graph section.

- Click and drag the side or corner handles of the selection frame to resize it.

- Click anywhere in a frame to set this position as center of the selection frame.

- Double-click a step in the top and center frames to open its graph on the next lower hierarchy level.

- CTRL+double-click a step in the top and center frames to open its graph on the next lower hierarchy level in a new graph tab.

- Click the center or top frame while you have the bottom frame active to update the Graph Window with the higher-level graph of the clicked frame.

Explorer

The Explorer window displays the structure of the currently active recipe or building block in the form of a tree view. The elements are ordered by their location in the graphs, from top to bottom and left to right. In the tree view, the system applies dark blue highlighting and bold white typeface to the structure level displayed in the Graph Window and indicates the current focus by italics.

Graph elements that are based on an **Approved** source building block are marked by an icon (see "Marker Icons" in Vol. 1).

To use the tree view of the Explorer for navigation within your recipe or building block, proceed as follows:

- Single-click an element in the tree to open the graph that contains the element in the active Graph Window. The element itself is selected in the Graph Window.

- CTRL+single-click an element in the tree to open the graph that contains the element in a new graph tab. The element itself is selected in the Graph Window.

- Double-click an element in the tree to open its graph in the active graph tab.

- CTRL+double-click an element in the tree to open its graph in a new graph tab.

> **TIP**
>
> Please note that double-clicking a dummy element or a pre-configured element that cannot be edited triggers the same behavior as a single click, moving you to the next higher level that displays the step in the graph, which consequently updates the Graph Window.
> Single-clicking a recipe root node does not trigger the Graph Window to update, since there is no header view available for master recipes.

Setlist

The **Setlist** represents a pool of data objects you can use for building and configuring your recipes. You can fill the Setlist from the Universe (page 94), when you mark a data object for display in the Setlist. There are, however, two types of data objects whose display in the Setlist is not configurable through the Universe:

- The display of ERP BOM items solely depends on whether or not the master recipe is based on an ERP BOM (page 137).

- Capabilities are provided with the system and thus always available.

The following types of data objects are available as collapsible tables in the Setlist:

- Material
  Material parameters are only available for phase elements.

  > **TIP**
  > Please note that the number of local input and local output materials you can assign to a building block depends on the basic configuration of the building block.

- ERP BOM Item
  ERP BOM item parameters are only available for recipes that are based on an ERP BOM.

- Equipment Class
  Equipment class parameters are only available for phase elements.

  > **TIP**
  > Please note that the number of equipment classes you can assign to a building block depends on the basic configuration of the building block.

- Property Type
  Property type parameters are only available for phase elements.

  > **TIP**
  > Equipment classes and property types are maintained in the Data Manager - Equipment application and may thus be affected by changes made there.

- Work Center
  Work center parameters are only available for unit procedure elements.

■ Station

Station parameters are only available for operation elements.

> **TIP**
>
> Work centers and stations are maintained in the Data Manager - Work Center application and may thus be affected by changes made there.

■ Signature Privilege

Signature privilege parameters are only available for phase elements.

■ Capability

Capability parameters are provided with the system and not available for all hierarchy levels. The marker icons in the **Levels** column indicate to which elements you can assign a capability:

The capability can be assigned to unit procedures.

The capability can be assigned to operations.

■ Procedure

■ Unit Procedure

■ Operation

■ Phase

> **TIP**
>
> A Setlist table can contain more data columns than initially visible in the default layout. Use the horizontal scroll bar at the bottom of the Setlist window or resize the window to access this data.

The Setlist toolbar provides toggle buttons that determine the mode or position with which a Setlist object is added to the selected component in the active graph of the Graph Window.

> **TIPS**
>
> Make sure to have the relevant graph component selected in the Graph Window and the relevant buttons toggled before you double-click to add an object.
>
> Please note that Dispense unit procedures or operations are not suitable for use in device processing. For this reason, the system unselects and removes the Dispense option when the building block is drawn into a device master recipe.

Click the **Sequence** button to insert the building block in sequence at the selected component.

Click the **Selection branch** button to insert a selection branch in parallel to the selected component.

⊥ ⊤ Click the **Simultaneous branch** button to insert a simultaneous branch in parallel to the selected component.

⊥ ⊤ Click the **Join** button to join the open branches at the selected component. The button is only enabled if a join operation is possible.

⇄ ⇄ Click the **Replace** button to replace the component assigned to the currently selected graph step with the component you select from the Setlist. It performs a full replacement, which means that all configurations made with the process parameters of the replaced graph component are overwritten and thus lost.

⇵ ⇵ Click the **Smart replace** button to replace the component assigned to the currently selected graph step with the component you select from the Setlist. Contrary to the full replacement of the **Replace** action, the **Smart replace** action merges the data configured with the parameters of the new replacing building block into the data of the old replaced building block.

SMART REPLACE CONCEPTS

If the **Smart replace** action is used for replacing a building block that contains elements, such as an operation building block with its phases, the system uses the identifiers of the individual elements to determine if the elements of the replacing building block match the elements of the replaced building block.

- Matching elements are merged.

- Elements of a replacing (new) building block, for which the system cannot establish a match, are treated as new elements and added with all of their parameters.

- Elements of a replaced (old) building block, for which the system cannot establish a match, are deleted.

Figure 5: Basic principles of Smart replace

The merge strategies of the **Smart replace** action differ with respect to the individual parameter types.

■ **Process parameters**

  ■ The system uses all **locked** parameters of the new building block, thus overwriting the respective parameters of the old building block.

  ■ For all **non-locked** parameters of the new building block that also exist in the old building block, the system retains the values of the old building block.

> **TIP**
> Please note that the system retains locks that were set explicitly on the old building block after it was drawn into the graph, but removes **Frozen** locks that were drawn in with an **Approved** building block.

  ■ Additional process parameters or parameter attributes on the new building block, which do not exist on the old building block, are taken over with their values.

  ■ Process parameters or parameter attributes that only exist on the old building block are removed.
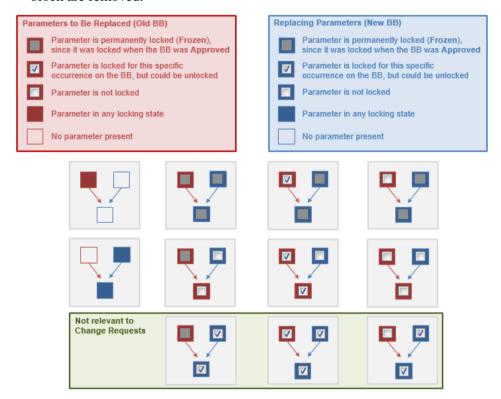
Figure 6: Smart replace for process parameters and their attributes

- **Equipment requirement**, **work center assignment**, **privilege**, and **capability parameters**

    - The system uses all **locked** parameters of the new building block, thus overwriting the respective parameters of the old building block.

    - For all **non-locked** parameters of the new building block that also exist in the old building block, the system retains the values of the old building block.

> **TIP**
> Please note that the system retains locks that were set explicitly on the old building block after it was drawn into the graph, but removes **Frozen** locks that were drawn in with an **Approved** building block.

    - Additional **non-locked** parameters on the new building block are ignored.

    - Parameters that only exist on the old building block are retained.
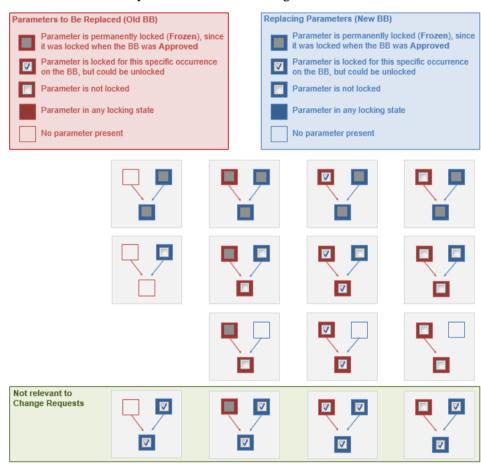
Figure 7: Smart replace for equipment requirement, work center assignment, privilege, and capability parameters and their attributes

■ **Material parameters**

    ■ The system ignores the material parameters of the new building block and instead retains all material parameters and MFC data of the old building block.

> **TIP**
> Structural differences in a new higher-level building block (operation or unit procedure), such as the deletion or insertion of a phase with MFC-relevant material parameters, affect the overall material flow.
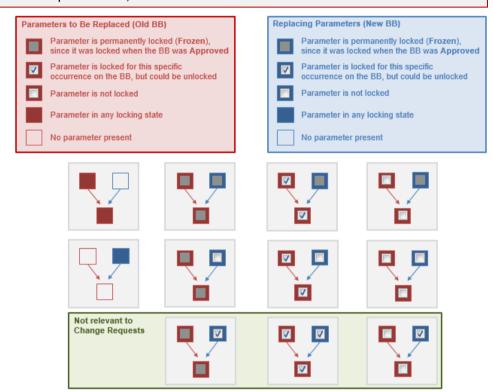
Figure 8: Smart replace for material parameters and their attributes

■ **Transitions**

- ■ The system uses all locked transitions of the new building block, thus overwriting the respective transitions of the old building block.

- ■ For all non-locked transitions of the new building block that also exist in the old building block, the system retains the values of the old building block.

> **TIP**
> Please note that the system retains locks that were set explicitly on the old building block after it was drawn into the graph, but removes **Frozen** locks that were drawn in with an **Approved** building block.

- ■ Additional transitions on the new building block, which do not exist on the old building block, are taken over with their values.

- ■ Transitions that only exist on the old building block are removed.
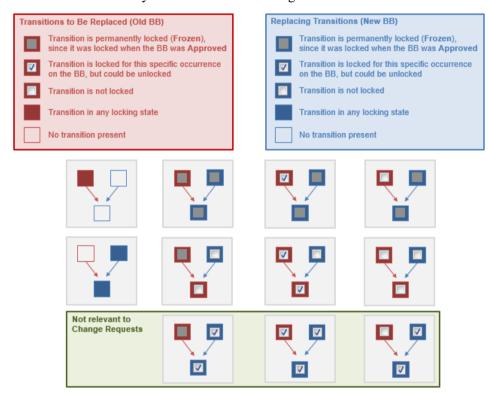
Figure 9: Smart replace for transitions and their attributes

Click the **Input** button to insert a material as input to the selected element.

Click the **Output** button to insert a material as output to the selected element.

Click the **Invert drawing direction** button to change the direction with which new elements are added to the graph, from down and rightwards to up and leftwards.

To edit the active graph, proceed as follows:

1. Toggle the relevant buttons to define the insertion rules for the object you are about to select. Which buttons are enabled, depends on the type and graph environment of the component you have selected. Initially, the **Sequence** and **Input** buttons are toggled by default.

2. In the Setlist, double-click the object you wish to add to the selected graph element.
   Depending on the object type, the system inserts new graph elements or parameters.

3. Double-click another Setlist object to add it in the same manner.
   The button settings remain in place until you actively click another button in the Setlist toolbar thus redefining the rules for adding Setlist objects.

Property Windows

You can access the properties of the recipe or building block headers (page 45) and elements (page 52), as well as the source (page 53) building blocks on which they are based from their respective property windows.

Header Property Window

The **Header** window lists the specific properties and the status of the component currently displayed in the Graph Window:

- a master recipe (page 45),

- the root component of a building block (page 50),

- or a change request (page 48).

Master Recipe Header

The **Header** window for master recipes lists the specific properties and the status of the master recipe that is currently displayed in the Graph Window.
The extra pane at the bottom shows the description of the master recipe, if this property is defined.

Click a property value to edit it. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

TIP

The **Identifier**, **Version**, **Access privilege**, and **Status Data** properties are never editable.

- To rename a master recipe, open the **File** menu and select the **Rename <component identifier>** function.

- To change the status of a master recipe, open the **File** menu and select the **Change status of <component identifier>** function.

- To create a new version of a master recipe, open the **File** menu, click the **Change status** button (see "Toolbar Buttons" in Vol. 1) in the **Status toolbar** or open the **File** menu and select the **Change status of <component identifier>** function.

The following properties are available for master recipes:

- **Identifier**

  Defined on the **New Master Recipe** dialog at master recipe creation.

- **Version**

  Set by the system at master recipe creation.

- **Description**

- **Comparison baseline**

  Displays the baseline master recipe against which the system compares the current master recipe and displays the comparison result in the Comparison window (page 67).

  > **TIP**
  > Please note that your access rights determine which master recipes are available for selection. Confidential master recipes that you are not allowed to access are not shown in the list.

  The recipe is shown with its identifier and version, which is displayed as **<major number>.x** as the version of a valid recipe may change at the minor number, for example when its status is changed from **Valid** to **Valid** in order to adjust its period of validity.

  To open the **Select Comparison Baseline** dialog (page 84) and assign a master recipe as baseline, click the **Open cell editor** button (see "Dialog Buttons" in Vol. 1).

  To delete the assignment, click the **Delete cell content** button (see "Dialog Buttons" in Vol. 1).

- **Method**

  Very often master recipes are maintained with different methods, which means there are several master recipe objects that only differ with respect to their method. However, in order to allow them to be valid at the same time, a given method must also be reflected within the master recipe identifier. Different versions of the same master recipe identifier cannot be valid at the same time. The maximum number of allowed characters in this box is 4.

- **Method description**

- **Reason for creation**

- **Access privilege**

  For confidential master recipes that require specific protection. It defines which access right a user needs to have to be able to access the master recipe and all orders that are based on it.

- **Status data**

  It comprises the master recipe's current status and period of validity.

■ Planned quantity

This value represents the reference quantity on the basis of which the system will calculate the material quantities when it explodes an order for the master recipe. The planned quantity of a device master recipe is always 1 and cannot be changed. Its unit of measure is derived from the material to be produced with the master recipe and is thus either **ea** or a unit convertible into **ea**.

■ Minimum quantity

■ Maximum quantity

■ Registration number

The maximum number of allowed characters in this box is 20.

■ Review mode

Specifies how an order that is based on the master recipe is moved to the **Reviewed** status in the Exception Dashboard of PharmaSuite for Production Responses.

　■ Automatic

　The status of the order automatically moves to **Reviewed** when it is **Finished** and no quality exceptions have been recorded for it or any of its appended workflows while it was being processed and if all of its appended workflows are either **Finished**, **Production-reviewed**, or **Reviewed**.

　■ Manual

　The status change to **Reviewed** needs to be performed manually when the order is **Finished** and there are no exceptions left that still need to be reviewed.

■ Comment

■ Material (product) data

It comprises the identifier and description of the material to be produced with the master recipe and additional information on its ERP BOM, if available.

> **TIP**
> The material to be produced with a master recipe is selected when the master recipe is created. You can change the material later with the **Replace material (product)** function from the File menu.

■ Packaging levels

Define the packaging structure of the material to be produced with the master recipe with up to four levels. The levels represent the layers of packaging required for the material. Each level is identified by its meaning, such as **Blister** or **Shipping carton**. Along with its meaning, it is possible to define the number of units (of the next lower level) the level contains. You can also set an **Inventory level** for each level to define if the level represents a **Sublot**, a **Logistic unit**, or both **Sublot and logistic unit** when handled in a warehouse context. The defined

data is available for quantity calculations and checks during execution. However, if the explicit level definitions are not of interest to an operator processing the material, selecting the **Hide during execution** option prevents the level settings from displaying in the respective phase.

> **TIP**
> Please note that the **Meaning** and **Contained number** attributes defined in the master recipe header must be identical to the respective settings made for the output material of the ERP BOM if an ERP BOM is used.

The system provides the Packaging Level Data editor for defining the data.

Change Request Header

The **Header** window for change requests lists the specific properties and the status of the change request that is currently displayed in the Graph Window.
The extra pane at the bottom shows the description of the change request, if this property is defined.

Click a property value to edit it. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

> **TIP**
> The **Identifier**, **Status Data**, and **Additional Data** properties are never editable.
>
> ■ To rename a change request, open the **File** menu and select the **Rename <component identifier>** function.
>
> ■ To change the status of a change request, click the **Change status** button (see "Toolbar Buttons" in Vol. 1) in the **Status toolbar** or open the **File** menu and select the **Change status of <component identifier>** function.

The following properties are available for change requests:

■ Identifier
Defined on the **New Change Request** dialog at change request creation.

■ Description

■ Comparison baseline handling
Defines which comparison baseline will be set for the new versions of the master recipes or building blocks:

  ■ **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, or

■ **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.

■ Target status
Indicates the status that the new versions of the master recipes are to have after the change request has been executed (**Edit**, **Verification**, **All valid statuses** comprising **Scheduled** and **Valid**).
If your change request will not include master recipes, but only building blocks, you can leave the property blank.

■ Effective date
Indicates the date when the changes introduced by the change request become effective. This means that when you execute a change request on a master recipe and define an effective date for it, the system behaves as follows:

■ If the effective date falls into the validity period of a valid master recipe, it keeps the valid version of the master recipe in the **Valid** status, but changes its period of validity to end at 11:59 PM on the day before the effective date.

■ If the effective date falls into the validity period of a scheduled master recipe, it keeps the unchanged scheduled version of the master recipe in the **Scheduled** status, but changes its period of validity to end at 11:59 PM on the day before the effective date.

■ It creates a version of the master recipe in the **Scheduled** status starting its validity on the day of the effective date.

■ On the day specified as effective date at 12 AM, the system moves the new, changed version of the master recipe to **Valid** and the old version to **Archived**.

> **TIP**
> If you do not define a recipe effective date for your change request, the system assumes that the changes need to become effective immediately. Consequently, on each affected master recipe, it sets the effective date to the current date (today) as soon as it starts to process the master recipe and executes the change request on it.
> Thus the new versions of the changed master recipes are created in the **Valid** status and immediately effective.
> The recipe effective date is not relevant to building blocks.

■ Status

■ Additional data
It comprises the structure level, identifier, revision, and access privilege of the building blocks selected for the change request.

Building Block Header

The **Header** window for building blocks lists the specific properties and the status of the root component of the building block that is currently displayed in the Graph Window. The extra pane at the bottom shows the description of the building block, if this property is defined.

Click a property value to edit it. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

> **TIP**
>
> The **Identifier**, **Revision**, **Access privilege**, and **Status Data** properties are never editable.
>
> ■ To rename the root component of a building block, open the **File** menu and select the **Rename <component identifier>** function.
>
> ■ To change the status of a building block, click the **Change status** button (see "Toolbar Buttons" in Vol. 1) in the **Status toolbar** or open the **File** menu and select the **Change status of <component identifier>** function.
>
> ■ To create a new revision of a building block, open the **File** menu, select the **Save as** function, and save the component under the same identifier.

The following properties are available for root components of building blocks:

> **TIP**
> Please note that you need to select the header element of the building block in the Graph Window before you can make changes to its header properties.

■ Identifier
Defined on the **New <Building Block Level>** dialog at building block creation.

■ Revision
Defined on the **New <Building Block Level>** dialog at building block creation.

> **TIP**
> If your building blocks will be subject to automatic creation of new revisions through change requests, it is recommended that revision numbers contain at least one numeric integer value. When the system creates a new revision of a building block while executing a change request, it increases the last integer value of a revision number to the next available integer.
> Examples:
> [1] -> [2]
> [A] -> [3] if revisions [1] and [2] exist already.
> [1.0] -> [1.1]
> [2B] -> [3B]
> [2B1] -> [2B2]
> [1B] -> [4B] if revisions [2B] and [3B] exist already.

■ Short description

■ Comparison baseline

Displays the baseline building block against which the system compares the current building block and displays the comparison result in the Comparison window (page 67).

> **TIP**
> Please note that your access rights determine which building blocks are available for selection. Confidential building blocks that you are not allowed to access are not shown in the list.

The building block is shown with its identifier and revision.

To open the **Select Comparison Baseline** dialog (page 84) and assign a building block as baseline, click the **Open cell editor** button (see "Dialog Buttons" in Vol. 1).

To delete the assignment, click the **Delete cell content** button (see "Dialog Buttons" in Vol. 1).

■ Access privilege

For confidential building blocks that require specific protection. It defines which access right a user needs to have to be able to access the building block.

> **TIP**
> Please note that in hierarchical structures, the required access privilege is determined by the component on the highest level of the structure.
> Example:
> When a phase building block with access privilege A is located in an operation that has access right B, a user who has access right B can open the operation and access the phase building block, but cannot open the phase building block directly.

■ Categories 1-3

Configurable filter categories.

■ Description

■ Dispense option

Only available for unit procedure and operation building blocks.

The **Dispense** option that marks an operation (see "Operation Header" in Vol. 1) or unit procedure (see "Unit Procedure Header" in Vol. 1) as Dispense-specific is only editable for operation building blocks. A unit procedure automatically displays the setting when it contains an operation for which the **Dispense** option has been set.

By marking an operation as Dispense-specific, you place the following restrictions on its usage in a unit procedure graph:

■ There can only be one operation that is marked as Dispense-specific within the graph.

■ A Dispense operation cannot be combined with other non-Dispense operations.

This means that a Dispense unit procedure must contain exactly one Dispense operation.

> **TIP**
> Please note that Dispense unit procedures or operations are not suitable for use in device processing. For this reason, the system unselects and removes the Dispense option when the building block is drawn into a device master recipe.

■ Status data
It comprises the building block's current status, its approval date and signature, and its archival date and signature, if available.

Element Property Window

The **Element** window lists the specific properties of the component selected in the Graph Window.
The extra pane at the bottom displays the full description of the element, if available.
The **Identifier** and **Description** properties are common to all components whereas the **Dispense** property is only available with unit procedures and operations.
The **Dispense** option is only visible for building blocks that contain a Dispense operation or unit procedure as element.
It is editable for operations (page 146) and must be set if the operation contains Dispense phases to form a default Dispense operation. It is read-only for unit procedures (page 144), which automatically receive the setting when they contain an operation for which the option hast been set.

> **TIP**
> Please note that Dispense unit procedures or operations are not suitable for use in device processing. For this reason, the system unselects and removes the Dispense option when the building block is drawn into a device master recipe.

Click a property value to edit it. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

> **TIP**
> The **Element** window is empty when you have a component **Header** tab displayed in the Graph Window or a dummy element selected.

Source Property Window

The **Source** window lists the properties of the building block on which the recipe or building block element (page 52) is based. The building block properties can only be viewed.

The System Source properties of a phase indicate important characteristics that are relevant to where and how the phase can be used.

■ Usable in
Indicates if the phase is available for use in **Master recipes**, **Master workflows**, or both.

■ Trigger
Indicates that the phase is a trigger phase and can thus be referenced in the **Trigger-enabled** capability (page 109) of an operation.

■ Server-run
Indicates that the phase is intended for being run on a server without user interaction. It needs to be located in an operation that holds the **Server-run** capability (page 109).

■ Pause-aware
Indicates that the phase can interpret a pause signal it receives from its unit procedure and is thus especially suited for use in a unit procedure that holds the **Pause-enabled** capability (page 109).

Phase Preview

The **Phase Preview** window displays how a phase is rendered in PharmaSuite for Production Execution when the recipe is processed on the shop floor. Since a dummy phase is empty, it also displays an empty preview.

> **TIP**
>
> Please note that some phases may rely on information or input that is not yet available when the preview is created. The preview of these phases only shows a corresponding system message.

Messages Window

When you edit master recipes or building blocks Recipe Designer concurrently checks your work for errors, warnings, or information you may require and displays them in the Messages window.

The Messages window consists of the window frame with a tool panel for filtering and navigation and two main areas:

- The tree panel (page 58) that displays a tree view with all nodes of the master recipe or building block for which the checks have returned messages.

- The list panel (page 58) that displays a list of all messages that have been found for the master recipe or building block and from which you can update the Graph Window to access and address the element indicated by the message.

The panels are divided by a slidable separator and can thus be freely resized, if necessary.

The tool panel provides toggle buttons to filter the display in both the tree and the list panels by message type (error, warning, or information). By default, all message types are toggled on.

Click the **Toggle error messages** button to toggle the display of error messages.

Click the **Toggle warning messages** button to toggle the display of warning messages.

Click the **Toggle information messages** button to toggle the display of information messages.

To the right of each toggle button, the system displays the number of messages of this type that is currently present in the active recipe or building block. This number triplet is also shown on the tab title of the Messages window.

In addition to the filter toggle buttons, the tool panel contains a breadcrumb trail that indicates the path to the element that is currently selected in the tree and thus highlighted in the list panel.

> **TIP**
>
> Please note that the breadcrumbs, the tree, and the list panel are synchronized by selection. This means that when you select a breadcrumb, a tree node, or a message in the list, the system updates the display in the other two Messages window components accordingly.

Messages Window - Tree Panel

Similar to the Explorer (page 34), the tree panel of the Messages window displays the structure of the currently active recipe or building block. It only displays those nodes, however, for which the continuous checks the system performs in the background have returned error, warning, or information messages.
Nodes that are directly affected by one or more check messages are represented in red font. Black font is used for nodes that are not directly affected but have affected child nodes.

Click a step in the tree to highlight all of its messages in the list panel (page 58). The breadcrumbs update accordingly.

Messages Window - List Panel

The list panel displays all messages the system returns for the currently active master recipe or building block. Each message has an identifier that is unique to the specific message or message category and indicates its type (error, warning, information) by an icon. The types of the messages, which the system returns, determine for a master recipe whether you can run the **Prepare for status change** function (page 17) on it and for a building block if you can change its status (see "Changing the Status of Building Blocks" in Vol. 1) to **Approved**:

⊗ Error
A master recipe with errors cannot be prepared for status change.
A building bock with errors cannot be approved.
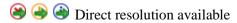
⚠ Warning
A master recipe with warnings cannot be prepared for status change.
A building bock with warnings can be approved, since the warnings can only be resolved after the building block has been integrated in a master recipe.

ⓘ Information
Messages of the information type do not affect the availability of the functions for status change of master recipes or building blocks.

🔴🟢🔵 Direct resolution available
For the error, warning, or information message, a direct solution is available by right-clicking the message to open a corresponding shortcut menu.

The messages in the list panel are grouped by the recipe or building block element for which they have occurred and are sorted in the same order as the elements are shown in the tree panel (page 58). All messages pertaining to the element selected in the tree are highlighted in gray.

The following actions are available from the list panel:

■ Single-click a message in the list to select the element for which it has occurred in the tree. All other messages pertaining to the element are highlighted in the list, along with the one you have clicked. The breadcrumbs update accordingly.

■ Double-click a message in the list to display the element to which the message refers.

   ■ If the message refers to a graph element, the system shows it in the Graph Window.

   ■ If the message refers to a parameter, the system also opens the Parameter Panel of the element in question and shows the affected parameter as selected.

   ■ If the message refers to an expression, the system also opens the Expression editor with the affected expression.

   ■ If the message refers to a property, the system selects it in its respective Property Window.

   ■ If the message refers to MFC data, the system opens the MFC tab.

■ CTRL+double-click a message in the list to display the graph element to which the message refers in a new tab of the Graph Window.

■ Right-click a message marked with a **Direct resolution available** icon to open the **Resolve issue** shortcut menu.

Messages Window - Details

For messages that either have complex causes and require actions or for which there is additional information available, the system provides the necessary information in the context-sensitive help system.
To access the relevant help topic, select the message in the list and press the F1 key.

The following messages provide additional information:

0001RS: The <...> is a dummy element.

The system provides dummy components on all levels of the recipe or building block hierarchy (dummy procedure (page 156), unit procedure (page 157), operation (page 156), and phase (page 156)). Their purpose is to help you with building the structure of your recipe or building block.

0002RS: The <...> contains a selection branch. Selection branches are not allowed on this structure level of the graph.

Only the graph of phases on operation and unit procedure level may contain branches and loops. The system does not support branches or loops between unit procedures.

0008RS: In the (...), a usage type is assigned to more than one privilege parameter.

Each usage type can only be assigned once per phase. In the Parameter Panel, the system highlights the privilege parameters with the non-unique assignments.

0009RS: The (...) has privilege parameter(s) with an exception risk assignment even though it does not support exceptions.

Whether a phase has pre-defined exceptions or not is controlled by the phase itself. If a phase has no exceptions pre-defined, the only privilege that you can configure is of the **Phase completion** usage type.

0010RS: The (...) has too few material input/output parameters (...).

The minimum and maximum numbers of material input and output parameters are defined with each phase building block. A phase is only configured correctly if the number of defined material parameters matches the pre-defined range.

0013RS: The start step and the end step are on different simultaneous branches.

This is a graph error that makes the graph invalid by SFC rules.
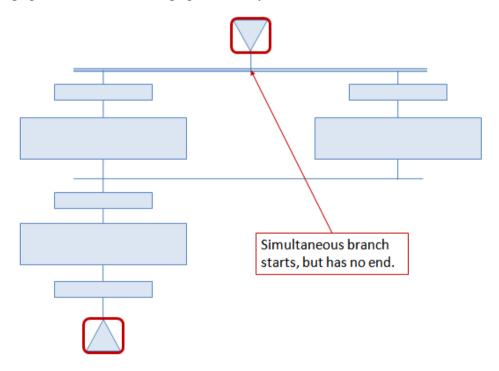
Figure 10: Example of graph error 0013RS

0014RS: A simultaneous branch cannot end as selection branch or loop.

This is a graph error that makes the graph invalid by SFC rules.
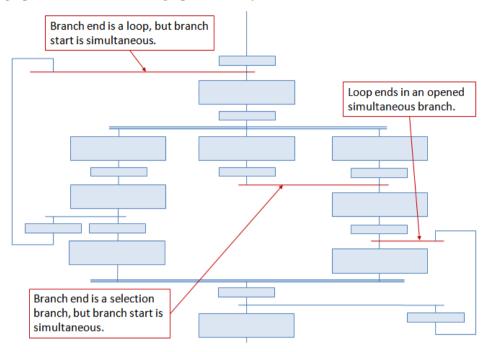
Figure 11: Example of graph error 0014RS

0015RS: A selection branch cannot end as simultaneous branch.

This is a graph error that makes the graph invalid by SFC rules.
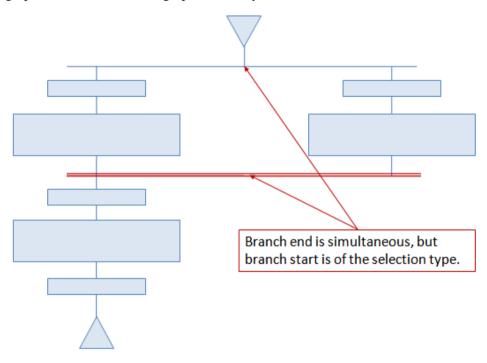
Figure 12: Example of graph error 0015RS

0017RS: The end of the simultaneous branch connects unsuitable components.

This is a graph error that makes the graph invalid by SFC rules.
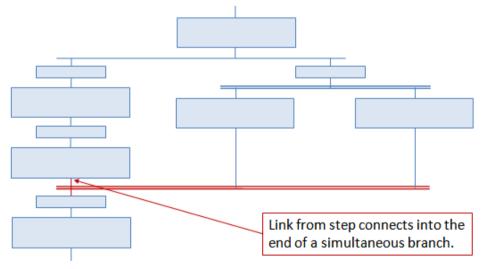
Figure 13: Example of graph error 0017RS

0025RS: The expression contains a reference to an unavailable element, which needs to be resolved.

When you use the **Create building block from selected** function to extract an element from an existing recipe, you may break references the building block may have had to other elements in its original recipe. In this case, the system prefixes the expression with a marker string (XT). When located in a building block, the issue is considered as warning, since it may not be possible to resolve it within the isolated structure of the building block. In a recipe, however, it is an error and must be resolved.

Thus when you draw a building block with the warning into a recipe, you have to remove the marker and check if you need to further adapt the reference.

TIP

When you lock the parameter or transition that contains the unresolved external reference, the warning turns into an error, thus preventing the building block from approval. Otherwise the entire parameter or transition would be frozen when drawn as element into a recipe or building block and could never be adapted to resolve the issue.
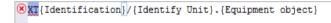
Figure 14: Expression with marker for external reference

0027RS: The transition starts a selection branch but has no condition expression defined.

During execution, the system has to decide which path through the branch the operator needs to take. For this reason each path must be preceded by a transition condition (page 151) that defines the circumstances under which the system will select the path for processing.
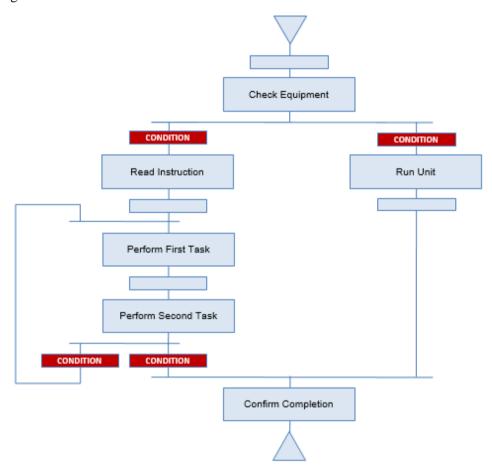
Figure 15: Sample graph with required transition conditions

0050RS: The procedure has no material input parameters to match the (...) position of the ERP BOM; required material: (...).

The system cannot match the ERP BOM position with the position of any of the material input parameters defined for the phases of your recipe. Make sure you have assigned a suitable material to a phase and have specified the same position for it as it has in the ERP BOM.

0051RS: The procedure has more than one material input parameter to match the (...) position of the ERP BOM.

The system has found more than one material input parameter whose positions are the same as the indicated position of the ERP BOM. Make sure you specify each ERP BOM position only once as position of a material input parameter in your recipe.

0055RS: The phase has an unsuitable material input parameter, since its (...) position does not correspond to the ERP BOM.

The system cannot match any ERP BOM position with the position of the material input parameter defined for the phase. Make sure you have specified the same position for it as it has in the ERP BOM.

0056RS: The phase has an unsuitable material input parameter, since its attributes do not correspond to those of the (...) position of the ERP BOM; required material: (...), quantity: (...), fixed quantity: (...).

The system cannot match one or all of the attributes of the ERP BOM item with the material input parameter defined for the phase. Make sure you have specified the same material with the same quantity as defined for the item in the ERP BOM.

0097RS: The Trigger-enabled capability contains a trigger that specifies an unavailable trigger phase and needs to be modified.

When you use the **Create building block from selected** function to extract an element from an existing recipe, you may break references the building block may have had to other elements in its original recipe. In this case, the system prefixes the expression with a marker string (XT). When located in a building block, the issue is considered as warning, since it may not be possible to resolve it within the isolated structure of the building block. In a recipe, however, it is an error and must be resolved.

Thus when you draw a building block with the warning into a recipe, you have to remove the marker and check if you need to further adapt the reference.

> **TIP**
>
> When you lock the capability that contains the unresolved external reference, the warning turns into an error, thus preventing the building block from approval. Otherwise the entire capability would be frozen when drawn as element into a recipe or building block and could never be adapted to resolve the issue.
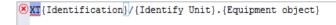

Figure 16: Expression with marker for external reference

Comparison Window

Recipe Designer supports you with comparing the master recipe or building block that is currently active in your Graph Window against a valid, approved, or archived baseline. As your currently active master recipe or building block can have any status, you can use the comparison function not only for approval purposes but also during editing, so that you can check your changes repeatedly while you edit.

The scope of the comparison has the following specific characteristics:

- When the comparison function detects a difference of the **Added object** or **Removed object** category, it stops the comparison for the object at this hierarchy level and does not attempt to match objects on lower levels.

- The comparison function only considers named components. So when you compare a recipe or building block that is work in progress and thus in the **Edit** status and contains unnamed, unconnected components, such as a lone branch or transition you forgot to delete after cutting it off from the graph, this difference is not detected. The system would, however report the invalid graph structure as error in the Messages window.

- When you compare a recipe or building block that does not show errors in its Messages window, all differences against its baseline are detected, regardless of the status of the recipe or building block.

The Comparison window consists of the window frame with a tool panel for filtering and navigation and two main areas:

- The tree panel (page 69) that displays a tree view with all nodes of the master recipe or building block that contain differences.

- The list panel (page 69) that displays a tabular listing of all differences that have been determined for the master recipe or building block and from which you can access the affected element for more information.

The panels are divided by a slidable separator and can thus be freely resized, if necessary.

The tool panel provides toggle buttons to filter the display in both the tree and the list panels by difference category (added, changed, or deleted). By default, all differences are toggled on.

Click the **Toggle added objects** button to toggle the display of all differences where an object was added.

Click the **Toggle changed objects** button to toggle the display of all differences where an object was changed.

Click the **Toggle removed objects** button to toggle the display of all differences where an object was removed.

To the right of each toggle button, the system displays the number of differences of this category that is currently present in the active recipe or building block. This number triplet is also shown on the tab title of the Comparison window.

For more refined filtering, the tool panel also holds the **Type** option list that allows you to filter for the following types of affected objects:

- Property (page 69)

- Successors (page 70)

- Preview sequence (page 71)

- Phase (page 71)

- Operation (page 73)

- Unit procedure (page 74)

- Procedure (page 75)

- Transition (page 76)

- Material parameter (page 77)

- Process parameter (page 77)

- Privilege parameter (page 78)

- Work center assignment (page 79)

- Equipment requirement (page 79)

- Capability (page 80)

- MFC (page 80)

In addition to the filter toggle buttons, the tool panel contains a breadcrumb trail that indicates the path to the element that is currently selected in the tree and thus highlighted in the list panel.

TIP

Please note that the breadcrumbs, the tree, and the list panel are synchronized by selection. This means that when you select a breadcrumb, a tree node, or a difference in the list, the system updates the display in the other two Comparison window components accordingly.

Comparison Window - Tree Panel

Similar to the Explorer (page 34), the tree panel of the Comparison window displays the structure of the currently active recipe or building block. It only displays those nodes, however, for which the comparison has detected differences. In addition to the nodes displayed in the Explorer tree, the Comparison tree also shows the nodes of named transitions as they may be detected to be different with respect to their identifiers, descriptions, or transition conditions (see "Expression Editor for Transition conditions" in Vol. 1).

Nodes that are directly affected by one or more differences are colored according to the difference category they contain:

- Added objects are green

- Changed objects are blue

- Deleted objects are red.

Black font is used for nodes that are not directly affected but have affected child nodes.

Click a step in the tree to highlight all of its messages in the list panel (page 69). The breadcrumbs update accordingly.

Comparison Window - List Panel

The list panel displays a tabular listing of all differences the system has detected between the currently active master recipe or building block and its comparison baseline. Each row of the table represents one difference, which is specified by its category.

⊕ Added object

⊛ Changed object

⊖ Removed object

Property

The difference type refers to the properties listed in the **Header**, **Element**, and **Source** property windows.

The following **Header** properties are excluded from the comparison:

- Identifier

- Version (only for master recipes)

- Revision (only for building blocks)

- Comparison baseline

- Status

- Valid from and Valid until (only for master recipes)

- Approved on and Approved by (only for building blocks)

- Archived on and Archived by (only for building blocks)

All **Element** properties are considered in the comparison.

The following **Source** properties in the **System Source** section are excluded from the comparison:

- Short description

- Usable in

- Trigger

- Server-run

- Pause-aware

EXAMPLE

	Type	Identifier	Attribute	New value	Old value
≠	Property	ID_SR-001	Minimum quantity	1000.0 g	

A **Changed object** difference has been detected for the **ID_SR-001** component.
Its **Minimum quantity** property (located in the **Header** property window) has been set to **1000.0 g**.
In the baseline, the property was not defined.

Successors

It indicates a difference in the graph when a named object (element or named transition) has a different successor. This can be caused by adding an element or a branch.

EXAMPLE

	Type	Identifier	New value	Old value
≠	Successors	Release Scale	Next Item, Compl./Released (Selection)	Back to ID, Compl./Released (Selection)

A **Changed object** difference has been detected for the **Release Scale** component.
It precedes a **Selection** branch with two transitions, **Next Item** and **Compl./Released**.
In the baseline, the first transition had a different identifier **Back to ID**.

Preview Sequence

It indicates a difference in the graph that causes the preview sequence of phases to be different during execution. This can be caused by changing the left-to-right sequence of the phases in a simultaneous branch.

EXAMPLE

	Type	New value	Old value
≠	Preview sequence	Trigger Equipment IPC Run, Trigger Material IPC Run	Trigger Material IPC Run, Trigger Equipment IPC Run

A **Changed object** difference has been detected, affecting the preview sequence of the **Trigger Equipment IPC Run** and **Trigger Material IPC Run** phases.
The **Trigger Equipment IPC Run** precedes the **Trigger Material IPC Run** phases.
In the baseline, it was the other way round.

Phase

It indicates a difference in the graph caused by a adding or removing a phase.
A **Phase** difference typically also means that there is a corresponding **Successor** difference.

> **TIP**
> Phases are compared by their identifiers, so renaming a phase results in two **Phase** differences, one of the **Added** category and one of the **Removed** category.
> Please be aware that the system does not try to match the data of **Added** objects, so any additional changes to the phase's properties or parameters would remain undetected.

EXAMPLE

	Type	Identifier	Predecessor	Successor
⊕	Phase	Manage Mixing Output	Start step, Release Scale (Selection)	Weigh, Completed (Selection)

An **Added object** difference has been detected for the **Manage Mixing Output** phase. It has the **Start step** and the **Release Scale** phase as predecessors from a **Selection** branch, which means it is located as first step in a loop. Its successors are two transitions (**Weigh**, **Completed**) that hold the conditions of a **Selection** branch that opens directly after the phase.
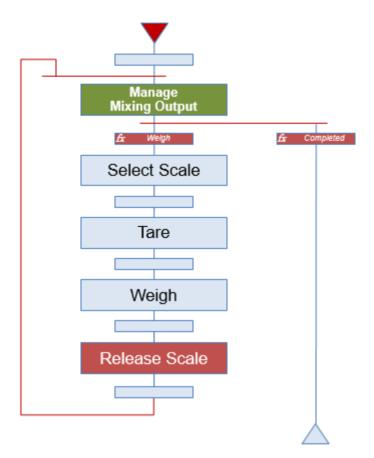
Figure 17: Phase difference - Added object

Operation

It indicates a difference in the graph caused by a adding or removing an operation.
An **Operation** difference typically also means that there is a corresponding **Successor** difference.

> **TIP**
>
> Operations are compared by their identifiers, so renaming an operation results in two **Operation** differences, one of the **Added** category and one of the **Removed** category.
> Please be aware that the system does not try to match the data of **Added** objects, so any additional changes to the operation's properties, parameters, or its phase graph would remain undetected.

EXAMPLE

	Type	Identifier	Predecessor	Successor
⊕	Operation	Weigh Output	Granulation, Sublot Preparation (Simultaneous)	End Step

An **Added object** difference has been detected for the **Weigh Output** operation.
It has the **Granulation** and the **Sublot Preparation** operations as predecessors from a **Simultaneous** branch, which means it is located as first step after the closing join of a simultaneous branch. Its successor is the **End Step**.
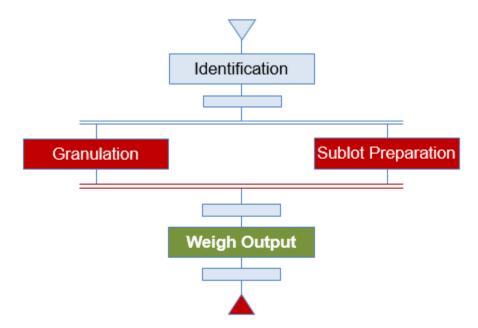
Figure 18: Operation difference - Added object

Unit Procedure

It indicates a difference in the graph caused by a adding or removing a unit procedure. A **Unit procedure** difference typically also means that there is a corresponding **Successor** difference.

> **TIP**
>
> Unit procedures are compared by their identifiers, so renaming a unit procedure results in two **Unit procedure** differences, one of the **Added** category and one of the **Removed** category.
> Please be aware that the system does not try to match the data of **Added** objects, so any additional changes to the unit procedure's properties, parameters, or its operation graph would remain undetected.

EXAMPLE

	Type	Identifier	Predecessor	Successor
⊕	Unit procedure	Granulating	Mixing	Tableting

An **Added object** difference has been detected for the **Granulating** unit procedure. It has the **Mixing** unit procedure as predecessor and the **Tableting** unit procedure as successor. This means it has been added in sequence between the two existing unit procedures.
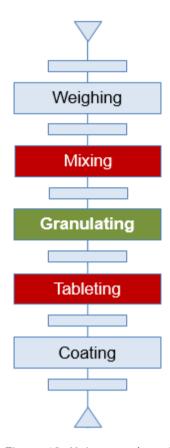
Figure 19: Unit procedure difference - Added object

Procedure

It indicates a difference in the graph caused by a adding or removing a procedure.
A **Procedure** difference typically also means that there is a corresponding **Successor** difference.

> **TIP**
> Procedures are compared by their identifiers, so renaming a procedure results in two **Procedure** differences, one of the **Added** category and one of the **Removed** category.
> Please be aware that the system does not try to match the data of **Added** objects, so any additional changes to the procedure's properties, parameters, or its unit procedure graph would remain undetected.

EXAMPLE

	Type	Identifier	Predecessor	Successor
⊕	Procedure	Sonolin Retard 80	Start step	End step
⊖	Procedure	Sonolin Retard 100	Start step	End step

An **Added object** difference has been detected for the **Sonolin Retard 80** procedure along with a **Removed object** difference for the **Sonolin Retard 100** unit procedure. This means that either the **Sonolin Retard 100** procedure has been replaced with the **Sonolin Retard 80** procedure or it has been renamed.

Transition

It indicates a difference in the graph caused by a adding or removing a transition. Since the system only regards named transitions as comparison-relevant, it reports a difference of the **Added** category when you define an identifier, a description, or an expression for an existing transition. Similarly, it reports a difference of the **Removed** category when you delete the identifier of a transition.

A **Transition** difference typically also means that there is a corresponding **Successor** difference.

> TIP
> Transitions are compared by their identifiers if they are named, and ignored if they do not have an identifier. So renaming an already named transition results in two **Transition** differences, one of the **Added** category and one of the **Removed** category.

EXAMPLE

	Type	Identifier	Attribute	New value	Old value
≠	Transition	Completed	Transition expression	={Preparation}/ {Identify Material}. {Identification result} == "COMPLETED"	={Preparation}/ {Weigh Charging Material}. {Weighing result} == "COMPLETED" OR {Preparation}/ {Identify Material}. {Identification result} == "COMPLETED"

A **Changed object** difference has been detected for the expression of the **Completed** transition.

To see the differences between the expressions, click either of the expressions in the **New value** or **Old value** column to open the **Differences** dialog.

Figure 20: Transition expression differences

Material Parameter

It indicates a difference in the Parameter Panel for material parameters. It can be of any category, as a material parameter can have been added, removed, or changed in one of its attributes.
A material parameter is identified by its position and material identifier.
The **Short description** attribute is excluded from the comparison.

EXAMPLE

	Type	Identifier	Attribute	New value	Old value
≠	Material parameter	60/D003-04	Default weighing method	Only identification	Net

A **Changed object** difference has been detected for the **Default weighing method** attribute of the material parameter for the **D003-04** material that has **60** as its position. The **Default weighing method** of the material parameter is now **Only identification** instead of **Net** as it was in the baseline.

Process Parameter

It indicates a difference in the Parameter Panel for process parameters. It can be of any category, as a process parameter can have been added, removed, or changed in one of its attributes.

The **Contents** attribute is excluded from the comparison.

	Type	Identifier	Attribute	New value	Old value
≠	Process parameter	Reprint label	Risk assessment	Low	High

A **Changed object** difference has been detected for the **Risk assessment** attribute of the **Reprint label** process parameter. The **Risk assessment** is now **Low** instead of **High** as it was in the baseline.

Privilege Parameter

It indicates a difference in the Parameter Panel for privilege parameters. It can be of any category, as a privilege parameter can have been added, removed, or changed in one of its attributes.
A privilege parameter is identified by its **Usage** attribute.

The following attributes are excluded from the comparison as they are maintained outside of Recipe Designer:

■ Reason (1st)

■ Reason (2nd)

	Type	Identifier	Attribute	New value	Old value
≠	Privilege parameter	Low risk exception	Usage name	Detach	

A **Changed object** difference has been detected for the **Usage name** attribute of the **Low risk exception** privilege parameter. The **Usage name** has been defined as **Detach** whereas in the baseline it was not defined.

Work Center Assignment

It indicates a difference in the Parameter Panel for work center assignments. It can only be of the **Added** or **Removed** category, as a work center assignment attributes are not editable in Recipe Designer.

The following attributes are excluded from the comparison as they are maintained in Data Manager:

- Description
- Type
- Storage area

Equipment Requirement

It indicates a difference in the Parameter Panel for equipment requirements. It can be of any category, as an equipment requirement can have been added, removed, or changed in one of its attributes.
A basic equipment requirement is identified by its parameter class. A complex equipment requirement that also includes a property type or a rule is identified by the identifiers of its parameter class and its property type or rule.

The following attributes are excluded from the comparison as they are maintained in Data Manager:

- Short description
- Description
- Equipment level

EXAMPLE

	Type	Identifier	Attribute	New value	Old value
≠	Equipment requirement	Coaters-A/ Drum Size	Rule	Drum Size: [30 .. 170] l [<>]	Drum Size: [50 .. 300] l [<>]

A **Changed object** difference has been detected for a **Rule** in the equipment requirement for the **Drum size** property type of the **Coaters-A** equipment class. The limits of its range are now at **[30 .. 170]** instead of **[50 .. 300]** as in the baseline.

Capability

It indicates a difference in the Parameter Panel for capabilities. It can be of any category, as a capability parameter can have been added, removed, or changed in one of its attributes.

A capability difference is identified by the capability's identifier and, if available, the affected action or trigger.

EXAMPLE

	Type	Identifier	Attribute	New value	Old value
≠	Capability	Trigger-enabled / Trigger-01	Trigger phase	{IPC Triggers}/ {Trigger Equipment IPC Run}	{IPC Triggers}/ {Trigger Material IPC Run}

A **Changed object** difference has been detected for the phase reference in the **Trigger-01** trigger of the **Trigger-enabled** capability.

To see the differences between the references, click either of the references in the **New value** or **Old value** column to open the **Differences** dialog.
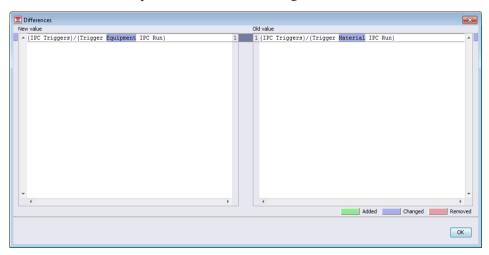
Figure 21: Capability reference differences

MFC

A difference of this type generally indicates that the MFC data of the recipe or procedure building block differs from the baseline. It opens the MFC tab, but does not mark further specifics.

The differences shown in the list panel are grouped by the recipe or building block component for which they have occurred and are sorted in the same order as the nodes are shown in the tree panel (page 69). The differences pertaining to the component selected in the tree are highlighted in gray.

The following actions are available from the list panel:

- Single-click a difference in the list to select the component for which it has occurred in the tree. All other differences pertaining to the component are highlighted in the list, along with the one you have clicked. The breadcrumbs update accordingly.

- Single-click the **Predecessor** or the **Successor** column of a difference to open the **Predecessor - Successor** dialog that juxtaposes the two component identifiers for better readability.
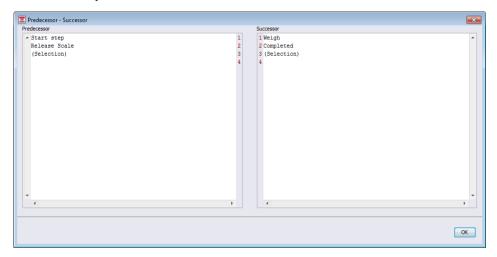

Figure 22: Comparison viewer: Predecessor - Successor

- Single-click the **New value** or the **Old value** column of a difference to open the **Differences** dialog that juxtaposes the two values and indicates their differences with color highlighting.
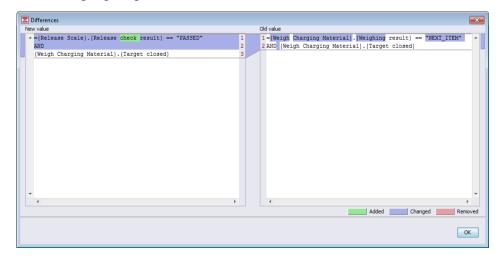
Figure 23: Comparison viewer: Differences

- Double-click a difference to jump to the object to which the difference refers.

  - If the difference refers to a graph element, the system shows it in the Graph Window.

  - If the difference refers to a parameter, the system also opens the Parameter Panel of the element in question and shows the affected parameter as selected.

  - If the difference refers to an expression, the system also opens the Expression editor with the affected expression.

  - If the difference refers to a property, the system selects it in its respective Property Window.

  - If the difference refers to MFC data, the system opens the MFC tab.

- CTRL+double-click a difference to display the graph element to which the message refers in a new tab of the Graph Window.

Status Bar

The status bar provides general information such as the logged-in user with his full name, the database name, and the local time and time zone information.

Dialogs

Recipe Designer uses dialogs or pop-up windows on top of its basic screen areas to cover a variety of functions:

- Open (page 83) and Select (page 84) dialogs for quick access to database objects to be used in Recipe Designer

- Dialogs of the print functions (page 85)

- Dialogs of status and version handling (page 87)

- Universe (page 94) for selecting data objects for the Setlist from the database

- Parameter Panel (page 95) for configuring the parameters of recipe or building block elements

- Statistics function (page 125) for master recipes and building blocks

- Context-sensitive Help system (page 127)

- About dialog (page 129) for further information on PharmaSuite and its system environment.

Open Dialog

The **Open** dialog supports you with choosing the recipe or building block you wish to open in the Graph Window.

Use the Search (page 13) and Filter (page 14) tools to restrict the number of items shown in the **Results** list.

The **Results** list displays all search and filter results and is updated along with each change to the search or filter criteria.

> **TIP**
>
> Please note that your access rights determine which master recipes or building blocks are available for selection. Confidential master recipes or building blocks that you are not allowed to access are not shown in the list.

If an object exists in several versions/revisions and statuses in the system, all of its occurrences are listed.
By default, the list is sorted alphabetically with respect to the **Identifier** column of the object, but you can adjust both the sort and column order (page 14).

■ Locate and double-click the data object you wish to open for use in Recipe Designer.
The system closes the dialog and displays the opened object in the Graph Window.

> **TIP**
>
> If the object you are trying to access is already open and thus locked in another session or by another user, the system allows you to open it in read-only mode and indicates this by the (R) marker on its upper tab title.

Select Dialog

The **Select** dialog supports you with

■ For choosing the material you wish to produce with a new master recipe (see "Toolbar Buttons" in Vol. 1).

> **TIP**
>
> Please note that the system only lists materials that are measured either in **ea** or a unit that is convertible into **ea**.

■ For choosing a material or ERP BOM to replace the current product material of an existing master recipe.

■ For choosing a comparison baseline against which the system can compare the current master recipe or building block and display the comparison result in der Comparison (page 67) window.

> **TIP**
>
> Please note that a master recipe must be **Scheduled**, **Valid**, or **Archived** to be available as comparison baseline.
> A building block must be **Approved** or **Archived**.

Use the Search (page 13) and Filter (page 14) tools to restrict the number of items shown in the **Results** list.

The **Results** list displays all search and filter results and is updated along with each change to the search or filter criteria.

> **TIP**
>
> Please note that your access rights determine which objects are available for selection. Confidential objects that you are not allowed to access are not shown in the list.

If an object exists in several versions/revisions and statuses in the system, all of its occurrences are listed.

By default, the list is sorted alphabetically with respect to the **Identifier** column of the object, but you can adjust both the sort and column order (page 14).

■ Locate and double-click the data object you wish to select for use in Recipe Designer.
The system closes the dialog and assigns the selected object to the master recipe.

Print Dialogs

Recipe Designer provides two functions to support you with printing a Master Recipe Report:

■ the **Open graph pagination** function available as toolbar button (see "Toolbar Buttons" in Vol. 1) or from the **File** menu (page 17), to display the tiled print preview of the graph that is currently active in the Graph Window and

■ the **Print report** functions available from the **File** menu (page 17), to open the print preview of the Master Recipe Report, which contains all relevant data of the master recipe along with graphical representations of the individual graphs for the master recipe's procedure and its subordinate structure levels.
You can either print the complete Master Recipe Report or print the report without the **Comparison with Baseline** section.
The functions are only available for saved master recipes.

On the report printout, graph representations are zoomed smaller than 100% to balance their print dimensions with readability. If a graph is still too wide or long to fit on one page, it is rendered in tiles.
You can view a graph's pagination at any time while you are building the graph so that you can monitor how it will look later in the printed report. This supports you with placing your graph components in the work area in a manner that keeps them from being cut on the printout.

> **TIP**
>
> Please note that you can only re-position graph components of editable graphs. Thus you cannot adjust the graph structure of **Valid** master recipes or **Approved** building blocks used in an otherwise editable master recipe or building block to improve their graphs' pagination.
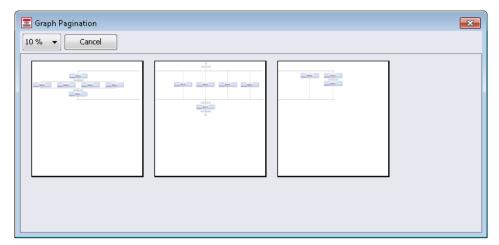
Figure 24: Graph Pagination dialog

The **Graph Pagination** dialog provides the following functions:

- To change the zoom factor of the displayed tiles, select one of the zoom options from the option list. The default zoom factor is 10%.

- To close the dialog, click the **Cancel** button.

> **TIP**
>
> Please note that the printed report always shows complete tiles for each graph view, even if a tile contains only one or two components.

Figure 25: Print Master Recipe Report (Device) dialog

The title of the **Print Master Recipe Report** dialog shows the identifier and version of the master recipe whose report preview it displ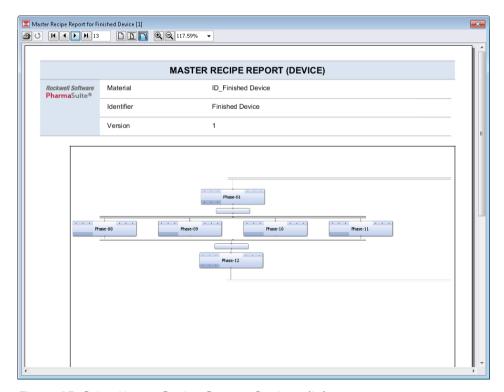ays. It provides standard print preview functions that allow you to print to a connected printer, page through the report, or resize and zoom the displayed pages.

Status Handling of Master Recipes

Master recipes and building blocks are maintained in Recipe Designer throughout their entire life-cycle. Their statuses are not affected by actions that take place in other applications of PharmaSuite. Thus all status changes are triggered from within Recipe Designer, either explicitly using the **Change status** function (page 88) or implicitly by executing a change request (page 166) that takes master recipes to the **Valid** status and building blocks to the **Approved** status.

Master Recipe Statuses

During their life-cycle, master recipes can assume the following statuses:

- **Edit**: The master recipe has been created and saved. It can be edited.

- **Verification**: The master recipe is ready for review. It is not editable any more. If updates are required, its status needs to be changed back to **Edit**.

- **Scheduled**: The master recipe is valid but its period of validity will start in the future.

- **Valid**: The master recipe is valid and available for use in order processing.

- **Archived**: The master recipe is no longer valid and cannot be used for order creation any more.

- **Obsolete**: The master recipe was never valid and will not be used for order creation or processing.

In addition to status control, master recipes are also subject to versioning (page 17). Changing the status of a master recipe can also affect its version number. In a typical configuration, a master recipe's version number would increase at the second digit, if its status were changed back from **Verification** to **Edit**.

> **TIP**
>
> Please note that only one version of a master recipe can be **Valid** at any given time. To extend the period of validity of a **Valid** master recipe, perform a status change from **Valid** to **Valid** and move the end date of its validity period into the future as required.

Depending on the system's configuration, a status change can require a user to enter a single or double electronic signature as proof of authorization.

Changing the Status of Master Recipes

To change the status of a master recipe, proceed as follows:

1. From the **File** menu, select **Change status of <master recipe identifier>** or click the **Change status** button (see "Toolbar Buttons" in Vol. 1) in the Status toolbar.

   > **TIP**
   > Make sure to have run the **Prepare <master recipe identifier> for status change** function before you try to move a master recipe out of the **Edit** status. Otherwise the **Change status of <master recipe identifier>** function is disabled.

   The system displays the **Change status** dialog for the master recipe that is currently active in the Graph Window. It displays the identifier, version, status, and description of the master recipe, along with its current validity dates.

2. From the option list, select the **Action** you wish to perform on the master recipe. The status (page 87) to which the action will move the master recipe is indicated in parentheses.

3. Use the Date Picker editor (see "Date/Time Picker Editor" in Vol. 1) to select the dates for the validity period, if required. The system presets the start date with the current date and the end date with 12/30/9999.

   > **TIP**
   > When selecting the dates, please note that 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 you can define a validity period of exactly one day by selecting the same date for both start and end dates.

   All **Available Versions** of the master recipe are listed in a sortable table (page 14) to support you with choosing the suitable action and validity dates. The list highlights the version whose status you are about to change.

   > **TIP**
   > Please note that changing a master recipe's status to **Valid** automatically moves the currently **Valid** master recipe to the **Archived** status.

4. Click the **OK** button.
   The system displays the **Electronic Signature** dialog. The **Description** box displays the status change you are about to perform.

5. Execute the electronic signature (page 15).

   - If the status change can be completed successfully, the **Change Status** dialog closes automatically.
     To view all status changes performed on the master recipe version, open its **Status History** window (page 90).

■ If the status change cannot be performed, the system indicates the issues in the **Versioning Conflicts** box and highlights the affected versions in the list of **Available Versions**.
Resolve the issues. Then re-click the **OK** button and complete the electronic signature.
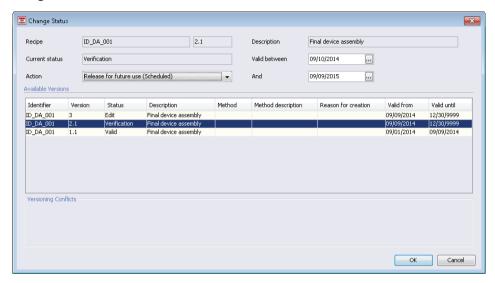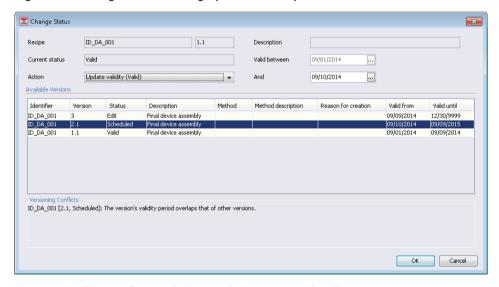

Figure 26: Change Status dialog of master recipe

Figure 27: Change Status dialog with Versioning Conflicts

Status History of Master Recipes

Once a master recipe has been saved for the first time, all of its status changes are registered and accessible for viewing at all times.

From the **File** menu, select the **View status history of <master recipe identifier>** function or click the **View status history** button (see "Toolbar Buttons" in Vol. 1) in the Status toolbar to open the **Status History** window of the master recipe that is currently active in the Graph Window. It displays the identifier, version, and status of the master recipe, along with its description and validity dates. The **History** of its status changes is listed in a sortable table (page 14) with the following data:

- **Timestamp** when the status change was performed.

- **Action** the user selected for the status change.

> **TIPS**
> Please note the specific handling of actions that applies to master recipes migrated from a pre-4.0 version of PharmaSuite.
> During the data migration process, actions are only inserted for status changes performed on the database on which the migration is being performed. Status changes recorded for the master recipe that were performed on a different database do not receive an entry in the **Action** column.
> If the underlying status and action model (Flexible State Model) of the master recipe is changed by a system administrator of PharmaSuite so that a status or action is no longer available, the system has to display the internal identifier of a missing action. To indicate this fact, the internal identifier is enclosed in asterisks, e.g. *verification*.

- **To status** indicates the status, to which the master recipe was moved.

- **To version** indicates the version the master recipe had after the status change.

> **TIP**
> Some status changes cause the version number to change as well.

- **Valid from** indicates the start date, time, and time zone of the master recipe's validity period.

> **TIP**
> Please note that the start date of the validity period includes the entire day and thus begins at 12:00 AM.

- **Valid to** indicates the end date, time, and time zone of the master recipe's validity period.

> **TIP**
> Please note that the end date of the validity period includes the entire day and thus lasts until 11:59 PM.

- **Signature** of the user who performed the status change, with full user name, login name, and timestamp.

- **Comment** the user could type on the signature dialog (page 15).

- **Database** on which the status change was performed.

> **TIP**
> This information is relevant if your environment uses different databases for recipe design and verification on the one hand and production on the other.
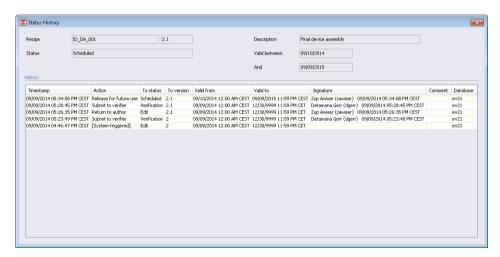

Figure 28: Status History dialog of master recipe

Status Handling of Change Requests

Change requests are subject to status management and are maintained within Recipe Designer throughout their entire life-cycle. Status changes to change requests are either triggered explicitly by user interaction through the **Change status** function (page 92) or automatically by the system when a change request is executed (page 166) and thus moves to another status.

Change Request Statuses

During their life-cycle, change requests can assume the following statuses:

- **Draft**: The change request has been created and saved. It can be edited.

- **In review**: The change request is ready for review. It is not editable any more. If updates are required, its status needs to be changed back to **Draft**.

- **Approved**: The change request is valid and available for execution.

- **In process**: The change request is currently being executed. The system automatically assigns this status to a change request as soon as a user has started its execution.

- **Finished**: The change request has been executed and has performed all changes. The system automatically assigns this status to a change request as soon as it has processed all changes listed in the change request's action list.

- **Archived**: The change request had either been processed to **Finished** or had become **Obsolete** and has now reached the end of its life-cycle.

- **Obsolete**: The change request has never been processed to **Finished**, but is no longer required.

- **Aborted**: The change request was canceled while it was being executed.

> **TIP**
>
> If an unexpected system shutdown occurs while a change request is **In process**, you cannot resume the change request after the system has been restarted. You can only manually move it either to **Obsolete** or to **Finished**, depending on what you consider more suitable in view of its progress. Additionally you need to release the locks the change request may still have on the objects it was processing.

Depending on the system's configuration, a status change can require a user to enter a single or double electronic signature as proof of authorization.

Changing the Status of Change Requests

To change the status of a change request, proceed as follows:

1. From the **File** menu, select **Change status of <change request identifier>** or click the **Change status** button (see "Toolbar Buttons" in Vol. 1) in the Status toolbar.
   The system displays the **Change status** dialog for the change request that is currently active in the Graph Window. It displays the identifier, status, and description of the change request.

2. From the option list, select the **Action** you wish to perform on the change request. The status (page 91) to which the action will move the change request is indicated in parentheses.

3. Click the **OK** button.
   The system displays the **Electronic Signature** dialog. The **Description** box displays the status change you are about to perform.

4. Execute the electronic signature (page 15).

   - If the status change can be completed successfully, the **Change Status** dialog closes automatically.
     To view all status changes performed on the change request version, open its **Status History** window (page 93).

■ If the status change cannot be performed, the system indicates the issues in the **Errors** box.
Click the **Cancel** button to close the **Change Status** dialog and resolve the issues.
Then re-open the **Change Status** dialog and perform the change.
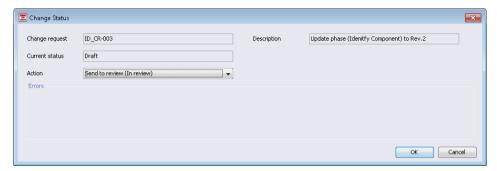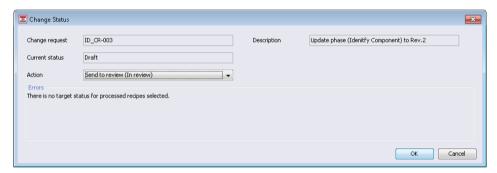

Figure 29: Change Status dialog of change request

Figure 30: Change Status dialog with Errors

Status History of Change Requests

Once a change request has been saved for the first time, all of its status changes are registered and accessible for viewing at all times.

From the **File** menu, select the **View status history of <change request identifier>** function or click the **View status history** button (see "Toolbar Buttons" in Vol. 1) in the Status toolbar to open the **Status History** dialog of the change request that is currently active in the Graph Window. It displays the identifier, description, and status of the change request. The **History** of its status changes is listed in a sortable table (page 14) with the following data:

■ **Timestamp** when the status change was performed.

■ **Action** the user selected for the status change.

> **TIP**
>
> If the underlying status and action model (Flexible State Model) of the change request is changed by a system administrator of PharmaSuite so that a status or action is no longer available, the system has to display the internal identifier of a missing action. To indicate this fact, the internal identifier is enclosed in asterisks, e.g. *verification*.

■ **To status** indicates the status, to which the change request was moved.

■ **Signature** of the user who performed the status change, with full user name, login name, and timestamp.

■ **Comment** the user could type on the signature dialog (page 15).
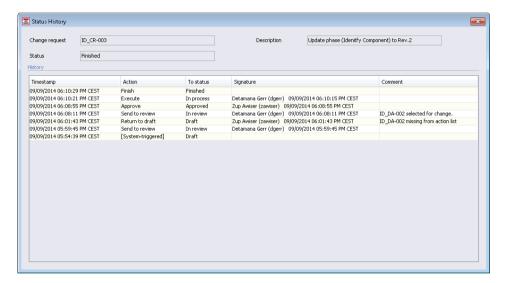

Figure 31: Status History dialog of change request

Universe

The **Universe** gives you direct access to relevant data objects stored in the database. From the Universe you can select the objects you wish to add to the Setlist (page 35) and subsequently use in your recipes or building blocks.

> **TIP**
>
> Please note that the Universe displays archived building blocks only for the purpose of using them as old building blocks when defining a change request (page 162) for mass changes. You cannot use archived building blocks for building recipes or building blocks.

Use the Search (page 13) and Filter (page 14) tools to restrict the number of items shown in the **Results** list.

The **Results** list displays all search and filter results and is updated along with each change to the search or filter criteria.

> **TIP**
>
> Please note that your access rights determine which objects are available for selection. Confidential objects that you are not allowed to access are not shown in the list.

By default, the list is sorted alphabetically with respect to the **Identifier** column of the object, but you can adjust both the sort and column order (page 14).

- In the **Setlist** column, select the checkbox of each object you wish to include in the Setlist (page 35) for later use in your recipe or building block. The system immediately updates the Setlist with your changes.

Parameter Panel

The **Parameter Panel** gives you access to all parameters of the selected element including the parameters of its child elements. The following types of parameters are available:

- Material parameters (page 95)
- Equipment requirements (page 101)
- Work center assignments (page 106)
- Privilege parameters (page 107)
- Capabilities (page 109)
- Process parameters (page 118)
- Transitions (page 124)

Material Parameters

Material parameters define which materials are required for the execution of a phase, as material inputs and outputs.

> **TIP**
>
> Please note that the number of local input and local output materials you can assign to a building block depends on the basic configuration of the building block.

When you open the Parameter Panel, it lists all parameters available at this level of your recipe or building block. Use the Search (page 13) and Filter (page 14) tools to restrict the number of items shown in the **Results** list.

By default, the parameters are sorted according to the hierarchy of their paths, but you can adjust both the sort and column order (page 14).
Table cells that contain editable values are displayed in black font, while read-only cells are shown in gray.
To set a parameter's values to read-only, select the checkbox in the **Lock** column. The box itself remains editable, so to revoke the setting, uncheck the box.

The **Results** list only displays the basic attributes of each parameter. The full list of attributes is shown in the **Attributes** panel to the right of the **Search** and **Filter** tools and the **Results** list. Editable basic attributes can be changed directly in the **Results** list as well as in the **Basic Data** section of the **Attributes** panel. Further attributes, such as weighing-specific attributes, are listed in additional sections.
The panels are divided by a slidable separator and can thus be freely resized, if necessary.

Please note the additional information on the following attributes:

- **Planned quantity mode**
  The planned quantity mode determines how the systems handles the planned quantity value during execution. Depending on the selected planned quantity mode of a material parameter, the system requires additional attributes to be set. Moreover, you cannot set all planned quantity modes for all material parameters.

  - **As defined**
    requires a planned quantity and tolerances to be defined.
    During execution, the system uses the value you define as planned quantity. It may, however, apply a prorate factor, for example to material inputs to maintain the intended ratio of materials.

  - **As produced**
    is not relevant to device recipes.
    It requires tolerances to be defined.
    During execution, the system draws the planned quantity from an output value of a preceding unit procedure, which means you can leave the planned quantity blank. If you have a planned quantity defined in the recipe, however, it is ignored.
    Thus **As produced** is only available for material transfers into a phase, but not for material inputs or outputs, since the referenced output value must be provided by a preceding phase.

■ **None**
is not relevant to device recipes.
The system does not expect a planned quantity value during execution, so you can leave the planned quantity blank. Even if you enter a value for the planned quantity or a value is downloaded from an external source, such as an ERP system, PharmaSuite ignores the value.
Only materials whose weighing material type is **Auxiliary substance** or not set (**---**), can have **None** as planned quantity mode.

■ **Planned quantity**
Whether or not you need to define a planned quantity, primarily depends on the **Planned quantity mode** of the material parameter. **As defined** parameters require a quantity to be set, whereas the attribute can remain blank for **As produced** or **None** parameters, since the system ignores it during execution.

■ **Position**
You have to define a position number for each MFC-relevant input parameter that is also an MFC input item. Position numbers must be unique throughout each execution path of a master recipe or building block. If you do not define the positions manually or import them from an ERP BOM (page 137), the system indicates in the Messages window (page 57) which input parameters lack their position numbers.
The positions are presented in any list of material inputs, for example during material identification on the shop floor, and are also used as default sorting criterion.

> **TIP**
>
> Weighing-specific attributes such as weighing method definitions or weighing material types only need to be maintained for material parameters used in weighing phases. Weighing method and tolerance values are drawn from the materials' master data, while weighing material types and a material's planned potency are preset with default values. You can change all preset values, even if they were initially drawn from the master data.

■ **Default weighing method**
Defines a default weighing method for the dispensing process, which is consequently pre-selected during execution. Depending on the allowed weighing methods, it can be overridden later on the shop-floor level during execution.

■ **Allowed weighing methods**
Defines which weighing methods are allowed for the material.
During execution, only the weighing methods that have been selected as allowed can be used for processing.
If the allowed weighing methods have been defined with the material data in

PharmaSuite for Production Management, they are populated automatically as parameter attributes to the relevant material parameter.

> **TIP**
> Allowing only one weighing method may lead to deadlock situations on the shop floor, when either Pallet weighing or Only identification have been set as only allowed weighing methods, but require a follow-up Net or Gross weighing to complete a weighing item in tolerance.
> Please note that the **Gross** weighing method covers both **Gross** and **Gross removal** weighing.

■ **Tolerances**
It is necessary to define at least one set of tolerances (absolute or relative), in order to support the dispensing process.
If both sets (absolute and relative) are maintained, the system will always use the minimum value of both. Nominal tolerances are calculated during order execution along with the calculation of the nominal quantities.

■ **Weighing material type**
The following calculation rules apply to the weighing of materials of the four weighing material types:

> **TIP**
> Please note that the calculations are only performed within a unit procedure, but not across unit procedures.

■ For active substances, the nominal quantity is calculated based on the actual potency of the identified batch.

■ For compensation substances, the nominal quantity is calculated based on the actual quantity of the active substance.

■ For filler substances, the nominal quantity is calculated based on the actual quantities of all BOM items that have been dispensed prior to the filler substance.

■ For auxiliary substances, the planned quantity is taken over as nominal quantity and does not require recalculation.

■ **Planned potency**
The planned potency is used as reference potency for the planned quantity. It is one of the input parameters for the calculation of the nominal quantity during the order execution. Potency-related calculations are only performed for materials of the **Active substance** weighing material type.

■ **Fixed quantity**
If you have a material input whose required quantity for processing does not scale with the quantity of material to be produced, select the **Fixed quantity** option. The planned quantity will then not be re-calculated during order explosion.

■ **Target weight**
During Output Weighing each produced sublot is checked against the defined target weight. For materials processed during Dispense and Inline Weighing processes, the attribute is not relevant.

■ **Target weight tolerances**
Define the tolerances for the pre-defined target weight. It is necessary to define at least one set of tolerances (absolute or relative), in order to support the check during execution. If both sets (absolute and relative) are maintained, the system will always use the minimum value of both.

■ **Packaging levels**
Define the packaging structure of the material with up to six levels. Level 0 indicates the basic dosage form of the material while levels 1 through 5 represent the layers of packaging required for the material. Each level is identified by its meaning, such as **Tablet** or **Vial** for level 0 and **Blister**, **Tray**, **Bundle**, or **Shipping carton** for levels 1 through 5. For each of the packaging levels 1 through 5, it is possible to define the number of units (of the next lower level) it contains. You can also set an **Inventory level** for each level to define if the level represents a **Sublot**, a **Logistic unit**, or both **Sublot and logistic unit** when handled in a warehouse context. The defined data is available for quantity calculations and checks during execution. However, if the explicit level definitions are not of interest to an operator processing the material, selecting the **Hide during execution** option prevents the level settings from displaying in the respective phase.

> **TIP**
> Please note that the **Meaning** and **Contained number** attributes defined with the material parameter must be identical to the respective settings made in the material's master data.

The system provides the Packaging Level Data editor for defining the data.

■ To edit the values of the parameter attributes, proceed as follows:

1. Click the table cell you wish to edit. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

2. Enter the value of the parameter attribute.

3. Press the ENTER key or click in another cell to conclude the edit action.

> **TIPS**
>
> Please be aware that changing MFC-relevant attributes, such as **Planned quantity**, **Position**, or the **MFC-relevant** option itself are reflected in the MFC tab of the Graph Window. Issues that may arise from changes of this kind are listed in the Messages window (page 57).
>
> Please note that **MFC-relevant** MFC output parameters are not allowed in device recipes.

◼ To lock one or more parameters, proceed as follows:

   1. Select or multi-select the parameters you wish to lock.

   2. Click the **Lock** button (see "Dialog Buttons" in Vol. 1).

◼ To unlock one or more parameters, proceed as follows:

> **TIP**
>
> Please note that you cannot unlock **Frozen** parameters.

   1. Select or multi-select the parameters you wish to unlock.

   2. Click the **Unlock** button (see "Dialog Buttons" in Vol. 1).

◼ To remove one or more parameters, proceed as follows:

> **TIP**
>
> Please note that you can only remove parameters that are not locked.

   1. Select or multi-select the parameters you wish to remove from the list of parameters displayed in the **Results** list.

   2. Click the **Remove parameter** button (see "Dialog Buttons" in Vol. 1).

> **TIP**
>
> When you edit a master recipe that is based on an ERP BOM (page 137), the Setlist provides all ERP BOM items with their relevant attributes (position, material identifier, description, and planned quantity) in a separate node. The attributes are populated to the respective cells in the Parameter Panel once a material parameter is selected from the **ERP BOM Item** node.
>
> A master recipe that is based on an ERP BOM must use each ERP BOM item as input parameter at least once. These input parameters must be marked as MFC-relevant.

Equipment Requirements

Equipment requirement parameters define the equipment requirements of a phase to specify exactly what kind of equipment is expected for use when a phase is processed in PharmaSuite for Production Execution. Equipment requirements can be equipment classes with or without additional properties or complex rules.
By adding properties to an equipment requirement you can either

- refine the rule of a dependent property whose property type already exists in the class or

- include an additional property that is independent of the requirement's class.

To define complex and more flexible equipment requirement rules that exceed the rule provided by a property type with pre-defined values, you can add flexible or conditional rules and specify their content in the Expression editor (see "Expression Editor" in Vol.1). With a flexible rule, you can define additional requirements that are not already covered by assigning a property type to the equipment requirement.
A conditional rule consists of two sections, which are introduced by commands (see "Expression Panel" in Vol.1). The first section defines a condition under which the rule will apply and the second section represents the rule itself. The result of a conditional rule must be boolean with the values **true** and **false**.

TIPS

Please note that conditional rules can be used with grouped equipment entities while flexible rules are only available for use with single equipment entities.

Please note that the number of equipment classes you can assign to a building block depends on the basic configuration of the building block.

When you open the Parameter Panel, it lists all parameters available at this level of your recipe or building block. Use the Search (page 13) and Filter (page 14) tools to restrict the number of items shown in the **Results** list.
In the list, each equipment parameter represents an equipment requirement and displays the data of the class that you have assigned to a phase to form the requirement. Further property types assigned to the requirement are shown in a sub-table under the requirement.

By default, the parameters are sorted according to the hierarchy of their paths, but you can adjust both the sort and column order (page 14).
Table cells that contain editable values are displayed in black font, while read-only cells are shown in gray.
To set a parameter's values to read-only, select the checkbox in the **Lock** column. The box itself remains editable, so to revoke the setting, uncheck the box.

> **TIP**
>
> Please note that the parameters must be locked to be replaced when you trigger the **Smart replace** action (see "Toolbar Buttons" in Vol. 1) from the Setlist toolbar.

- To edit the values of the parameter attributes, proceed as follows:

    1. Click the table cell you wish to edit. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

    2. Enter the value of the parameter attribute.

    3. Press the ENTER key or click in another cell to conclude the edit action.

- To lock one or more parameters, proceed as follows:

    1. Select or multi-select the parameters you wish to lock.

    2. Click the **Lock** button (see "Dialog Buttons" in Vol. 1).

- To unlock one or more parameters, proceed as follows:

    > **TIP**
    >
    > Please note that you cannot unlock **Frozen** parameters.

    1. Select or multi-select the parameters you wish to unlock.

    2. Click the **Unlock** button (see "Dialog Buttons" in Vol. 1).

> **TIP**
>
> During execution, an equipment entity is matched against the requirements defined with the recipe for which an operator plans to use it. Only if this validation check concludes that the properties of the equipment entity meet the requirements can the entity be used as planned.

Validation rules:

The rules that determine if an equipment entity matches its requirements take place on three levels of complexity:

- The basic level only has one rule, which checks if all properties defined for the requirement are also defined as properties of the equipment entity.

- The second level has a set of rules that apply if requirement and entity both state a value for a property, if one of them states a range while the other supplies a value, or if one or both of them state neither a value nor a range for a property.

Condition	Result
Requirement: value and range are not defined	The entity is a **valid** match for the requirement.
Requirement: value or range is defined **Entity:** value and range are not defined	The entity is **no valid** match for the requirement.
Requirement: value is defined (ReqValue) **Entity:** value is defined (EntValue) ReqValue = EntValue	The entity is a **valid** match for the requirement.
Requirement: range is defined (ReqLowLimit and/or ReqHighLimit) **Entity:** value is defined (EntValue) ReqLowLimit <= EntValue <= ReqHighLimit	The entity is a **valid** match for the requirement.
Requirement: value is defined (ReqValue) **Entity:** range is defined (EntLowLimit and/or EntHighLimit) EntLowLimit <= ReqValue <= EntHighLimit	The entity is a **valid** match for the requirement.
Requirement: value is not defined **Entity:** value is not defined **Either requirement or entity:** one value (Low or High) of range is defined	Covered by third level of rules, see below.
Either requirement or entity: value is defined range is defined (Low and/or High)	This condition represents an invalid configuration and is prevented by the system.

- ■ The set of rules on the third level applies to cases when both requirement and entity have a range defined for a property.
  When matching the range of an entity (EntLowLimit, EntHighLimit) against the range of a requirement (ReqLowLimit, ReqHighLimit) the system has to consider the following possible cases:

- Entity range is **lower** than requirement range:
  EntHighLimit < ReqLowLimit

- Entity range **overlaps** requirement range at the **low limit**:
  EntLowLimit < ReqLowLimit <= EntHighLimit <= ReqHighLimit

- Entity range **is contained** in requirement range:
  ReqLowLimit <= EntLowLimit <= EntHighLimit <= ReqHighLimit

- Entity range **contains** requirement range:
  EntLowLimit <= ReqLowLimit <= ReqHighLimit <= EntHighLimit

- Entity range **overlaps** requirement range at the **high limit**:
  ReqLowLimit <= EquLowLimit <= ReqHighLimit < EquHighLimit

- Entity range is **higher** than requirement range:
  ReqHighLimit < EntLowLimit

The following table lists how the system evaluates the settings made for entity and requirement ranges and indicates the match results.
If a range only has one limit, the system treats it as open range.

Limit settings specified				Range check type of requirement	
ReqLow	ReqHigh	EntLow	EntHigh	Is contained in entity range	Contains entity range
---	---	---	---	Valid match	Valid match
---	---	---	Yes	Valid match	Valid match
---	---	Yes	---	Valid match	Valid match
---	---	Yes	Yes	Valid match	Valid match
---	Yes	---	---	No valid match	No valid match
---	Yes	---	Yes	Valid match, if EntHigh <= ReqHigh	Valid match, if ReqHigh <= EntHigh
---	Yes	Yes	---	No valid match	No valid match
---	Yes	Yes	Yes	Valid match, if EntHigh <= ReqHigh	No valid match
Yes	---	---	---	No valid match	No valid match
Yes	---	---	Yes	No valid match	No valid match
Yes	---	Yes	---	Valid match, if ReqLow <= EntLow	Valid match, if EntLow <= ReqLow
Yes	---	Yes	Yes	Valid match, if ReqLow <= EntLow	No valid match
Yes	Yes	---	---	No valid match	No valid match

Limit settings specified				Range check type of requirement	
ReqLow	ReqHigh	EntLow	EntHigh	Is contained in entity range	Contains entity range
Yes	Yes	---	Yes	No valid match	Valid match, if Equ.max <= Req.max
Yes	Yes	Yes	---	No valid match	Valid match, if Req.min <= Equ.min
Yes	Yes	Yes	Yes	Valid match, if ReqLow <= EntLow **AND** EntHigh <= ReqHigh	Valid match, if EntLow <= ReqLow **AND** ReqHigh <= EntHigh

- ■ To add properties to equipment requirements in order to refine existing or add further properties, proceed as follows:

  1. Click the **Add ...** button (see "Dialog Buttons" in Vol. 1) of the requirement you wish to specify further with additional properties.
     The system opens an option list that holds the property types contained in the Setlist.

  > **TIP**
  > The system indicates dependent properties, which are already included in the class of the equipment requirement, with the **Dependent property** marker (see "Marker Icons" in Vol. 1).

  2. Select the property you wish to add to the requirement.
     The system displays the additional property in a sub-table under the requirement.

- ■ To add flexible or conditional rules to equipment requirements in order to refine existing or add further properties, proceed as follows:

  1. Click the **Add ...** button (see "Dialog Buttons" in Vol. 1) of the requirement you wish to specify further with a complex rule.
     The system opens an option list that holds the **Flexible rule** and Conditional Rule as first and second options.

  2. Select the rule option you wish to add to open the Expression editor (see "Expression Editor" in Vol.1) for equipment requirement rules.
     Define the rule and save it.
     The system displays the new rule in a sub-table under the requirement.

  > **TIP**
  > In the Parameter Panel, the identifier and description of a rule is not editable. To change it, you have to open the Expression editor and make your changes there in the respective input boxes.

■ To remove one or more parameters, proceed as follows:

> **TIP**
> Please note that you can only remove parameters that are not locked.

1. Select or multi-select the parameters you wish to remove from the list of parameters displayed in the **Results** list.

2. Click the **Remove parameter** button (see "Dialog Buttons" in Vol. 1).

Work Center Assignments

Work center assignments define where a recipe's phases are processed during execution:

■ per unit procedure, at which work center and

■ per operation, at which station of a work center.

You can assign one or more work centers to a unit procedure to specify that its operations have to be processed at one of the defined work centers.
Once you have specified the work center (or work centers) at which the unit procedure is available for processing, you can also define at which station (or stations) of the specified work center (or work centers) the unit procedure's operations may be processed.

Unless you make an explicit work center assignment for a unit procedure, its operations can be started and executed at any work center. Similarly, an operation of a unit procedure with work center assignment can be processed at any station of the specified work center, unless you have explicitly assigned it to one or more stations.

When you open the Parameter Panel, it lists all parameters available at this level of your recipe or building block. Use the Search (page 13) and Filter (page 14) tools to restrict the number of items shown in the **Results** list.

By default, the parameters are sorted according to the hierarchy of their paths, but you can adjust both the sort and column order (page 14).
Table cells that contain editable values are displayed in black font, while read-only cells are shown in gray.
To set a parameter's values to read-only, select the checkbox in the **Lock** column. The box itself remains editable, so to revoke the setting, uncheck the box.

■ To edit the values of the parameter attributes, proceed as follows:

1. Click the table cell you wish to edit. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

2. Enter the value of the parameter attribute.

3. Press the ENTER key or click in another cell to conclude the edit action.

■ To lock one or more parameters, proceed as follows:

    1. Select or multi-select the parameters you wish to lock.

    2. Click the **Lock** button (see "Dialog Buttons" in Vol. 1).

■ To unlock one or more parameters, proceed as follows:

> **TIP**
> Please note that you cannot unlock **Frozen** parameters.

    1. Select or multi-select the parameters you wish to unlock.

    2. Click the **Unlock** button (see "Dialog Buttons" in Vol. 1).

■ To remove one or more parameters, proceed as follows:

> **TIP**
> Please note that you can only remove parameters that are not locked.

    1. Select or multi-select the parameters you wish to remove from the list of parameters displayed in the **Results** list.

    2. Click the **Remove parameter** button (see "Dialog Buttons" in Vol. 1).

Privilege Parameters

Privilege parameters are used to control signature rights in the production environment. You can configure signature privileges for usage in

■ signatures that are requested at phase completion,

■ signatures for phase-specific exceptions, along with the risk level of the exception (high risk, medium risk, low risk, no risk).

> **TIP**
>
> Please note that only signatures of the **Phase completion** type are visible in the Phase Preview.
>
> During phase execution on the shop floor, however, some phases can suppress a phase completion signature even though it is configured in the relevant privilege parameter. This may be the case if a phase's behavior within a process automatically ensures the GMP-compliance of its usage so that halting the process for an operator to perform a signature is not necessary.
> The **Tare** phase of a Dispense operation provides an example for this behavior. It suppresses the phase completion signature during **Net** weighing, when the tare value is read automatically from the scale and cannot be influenced by operator actions. In **Gross** and **Pallet** weighing, however, when an operator manually enters tare values, the phase completion signature is displayed.

When you open the Parameter Panel, it lists all parameters available at this level of your recipe or building block. Use the Search (page 13) and Filter (page 14) tools to restrict the number of items shown in the **Results** list.

By default, the parameters are sorted according to the hierarchy of their paths, but you can adjust both the sort and column order (page 14).

Table cells that contain editable values are displayed in black font, while read-only cells are shown in gray.

To set a parameter's values to read-only, select the checkbox in the **Lock** column. The box itself remains editable, so to revoke the setting, uncheck the box.

■ To edit the values of the parameter attributes, proceed as follows:

1. Click the table cell you wish to edit. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

2. Enter the value of the parameter attribute.

3. Press the ENTER key or click in another cell to conclude the edit action.

> **TIP**
> Once you have assigned a signature privilege to your phase, operation, or unit procedure, you can decide if you wish to define the label for the login name and password boxes specifically for this occurrence of the signature or if you wish to use the default label defined with the signature privilege.
> **Reason (1st)** defines the label for a single signature or the first signature in a double signature and overrides the value of **Default reason (1st)**, which is drawn from the signature privilege.
> **Reason (2nd)** defines the label for the second signature in a double signature and overrides the value of **Default reason (2nd)**, which is drawn from the signature privilege.

■ To lock one or more parameters, proceed as follows:

1. Select or multi-select the parameters you wish to lock.

2. Click the **Lock** button (see "Dialog Buttons" in Vol. 1).

■ To unlock one or more parameters, proceed as follows:

> **TIP**
> Please note that you cannot unlock **Frozen** parameters.

1. Select or multi-select the parameters you wish to unlock.

2. Click the **Unlock** button (see "Dialog Buttons" in Vol. 1).

■ To remove one or more parameters, proceed as follows:

> **TIP**
> Please note that you can only remove parameters that are not locked.

1. Select or multi-select the parameters you wish to remove from the list of parameters displayed in the **Results** list.

2. Click the **Remove parameter** button (see "Dialog Buttons" in Vol. 1).

Capabilities

Capability parameters define if certain behaviors or capabilities are available during recipe execution.

PAUSE-ENABLED

The **Pause-enabled** capability is only available for unit procedures. It defines that during execution in PharmaSuite for Production Execution, processing can be paused for all phases that are pause-aware, which means they are capable of interpreting the pause signal they receive when their unit procedure is set to **Paused**.

The **Pause-enabled** capability is especially suited for use with operations that have the **Server-run** capability (page 109), such as operations that contain trigger phases for triggering runs of **Event-triggered** operations, since it holds their triggering activities. Thus it is possible to prevent operation runs from stacking during execution interruptions like shift changes or urgent equipment repairs.

> **TIPS**
>
> Please note that escalation scenarios of operation runs defined with the **Escalation-enabled** capability are not affected when an operator pauses a unit procedure.
>
> You can verify if a phase is able to interpret a pause signal from its unit procedure by opening its **Source** property window (page 53) and checking in the **System Source** section that the **Pause-aware** property is selected.

Action	Comment
Pause unit procedure	Controls if a unit procedure can be paused.

Attribute	Type	Comment
Exception enabled	Flag	Controls if the system tracks pausing a unit procedure as exception.
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

> **TIP**
>
> When you define an exception for the **Pause-enabled** capability you also need to define a signature privilege (page 107) for the unit procedure whose **Usage** attribute matches the **Risk assessment** of the exception.

⏏ DETACHABLE

The **Detachable** capability is available for both unit procedures and operations, but with the following restriction:

- Unit procedures may not contain operations that hold the **Event-triggered** capability.

It controls if an operator can detach a unit procedure from the work center or an operation from the station at which it is being processed so that its execution can be continued at another work center or station. You can define if detaching is treated as exception.
When a unit procedure is detached from its work center and the action is configured as exception, the system adds a detach exception to the unit procedure.
However, when an operation is detached from its station and the action is configured as exception, the system adds a detach exception to each active phase of the operation when the operator detaches the operation.

Action	Comment
Detach	For unit procedures: Controls if the unit procedure can be detached from its work center after it has been started. For operations: Controls if the operation can be detached from its station after it has been started.

Attribute	Type	Comment
Exception enabled	Flag	For unit procedures: Controls if the system tracks detaching the unit procedure as exception at the unit procedure. For operations: Controls if the system tracks detaching the operation as exception at the affected phases.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

TIP

When you define an exception for the **Detachable** capability you also need to define a signature privilege (page) for the unit procedure or operation, respectively, whose **Usage** attribute matches the **Risk assessment** of the detach exception.

◁ ABORT-AND-REACTIVATE-ENABLED

The **Abort-and-reactivate-enabled** capability is available for unit procedures. It controls if an operator can abort a unit procedure during processing and in the same action reactivate it so that it is available for being started again from the pool of unit procedures selectable for processing. This means that the reactivated unit procedure starts from scratch, with the first phase of its first operation.

The **Abort-and-reactivate-enabled** capability is especially suited for repair and rework situations, for example when an operator at a repair work center decides that a processing step needs to be redone completely.
When a unit procedure is aborted and reactivated and the action is configured as exception, the system adds a corresponding exception to the unit procedure.

Action	Comment
Abort and reactivate	Controls if the unit procedure can be aborted and reactivated so that it is available for reprocessing from scratch.

Attribute	Type	Comment
Exception enabled	Flag	Controls if the system tracks aborting and reactivating the unit procedure as exception at the unit procedure.

Attribute	Type	Comment
Risk assessment	Choice list	Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

TIP

When you define an exception for the **Abort-and-reactivate-enabled** capability you also need to define a signature privilege (page 107) for the unit procedure whose **Usage** attribute matches the **Risk assessment** of the exception.

EVENT-TRIGGERED

The **Event-triggered** capability is only available for operations that do not hold the **Auto-startable** capability. During processing, an event-triggered operation only exists as non-executable template until an external trigger or an operator action creates an executable run of the template.

Once a run has been created, an operator can start it like any other operation to process its phases. When the operator confirms the last phase of the run, the template remains available for creating further runs. It is also possible to create additional runs of the template while there are already startable runs or runs that are being processed.

As long as a template is displayed during processing and thus available for creating new runs, an operator cannot start succeeding operations for processing. For this reason, the system provides a function to remove the template, so that the operator can continue with processing.

The **Event-triggered** capability is especially suited for operations that need to be performed repeatedly at definable intervals, such as in-process controls (IPC).

Action	Comment
New run	Manually creates a new run from a template.
Cancel run	Only available for unstarted runs. Cancels the run.
Cancel all runs	Cancels all unstarted runs, while ignoring all started runs.

Action	Comment
Remove template	Removes the template from processing, thus finishing the operation.

Attribute	Type	Comment
Exception enabled	Flag	Available for all actions. Controls if the system tracks the respective action as exception.
Risk assessment	Choice list	Available for all actions. Defines the risk level of the exception and thus controls the related signature privilege. Available settings: **None, Low, Low (mandatory comment), Medium, Medium (mandatory comment), High, High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Available for all actions. Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

> **TIP**
>
> When you define an exception for the **Event-triggered** capability you also need to define a signature privilege (page 107) for the operation whose **Usage** attribute matches the **Risk assessment** of the exception.

TRIGGER-ENABLED

The **Trigger-enabled** capability is only available for operations that also hold the **Event-triggered** capability. It defines which phase or phases provide the external triggers that create an executable run of the operation. An operation can reference several trigger phases so that it can be activated by different events. This allows you to design in-process controls that are not only triggered at specific intervals but also when a defined count of produced goods has been reached. Trigger phases are typically located in operations that hold the **Server-run** capability (page 109).

When you reference a phase as trigger phase of your operation, you need to be aware of the following restrictions:

- The phase must be located within the same unit procedure as the operation it triggers.

- During processing, the phase must be active at the same time as the operation it triggers. This means that it has to be located

  - either in an operation that is on a simultaneous branch to the operation it triggers

  - or within the triggered operation itself where it would trigger runs of the operation once an operator on the shop floor has started the first run of the operation.

The trigger phase is typically but not necessarily a server-run phase.

TIP

You can verify if a phase is a trigger phase and thus available for referencing by opening its **Source** property window and checking in the **System Source** section that the **Trigger** property is selected.

Please note that both the **Time-based Trigger** and the **Counter-based Trigger** phases can only be used in operations that run on a server.

Attribute	Type	Comment
Trigger	Text	Defines the identifier of the trigger. It is preset with a system-generated suggestion. You can edit the identifier or type a completely different one. The maximum number of characters for a trigger is 40 and it must be unique within the operation.
Description	Text	Indicates the purpose of the trigger. It has only informational value.
Trigger phase	String	Reference to the trigger phase. Depending on the location of the trigger phase, the reference consists of either an operation identifier and a phase identifier or only a phase identifier. The identifiers are enclosed in curly brackets (braces) and separated by slashes.

For inserting references you can either type the texts yourself or make use of the intelligent auto-completion feature that opens automatically when you click in the **Trigger phase** cell, provided there are any phases suitable for referencing in your graph.

ESCALATION-ENABLED

The **Escalation-enabled** capability is only available for operations that also hold the **Event-triggered** capability. During processing, the system monitors the time that passes after a run of an event-triggered operation has been created, either automatically due to a trigger or manually by an operator. Once the run has been triggered and is thus available for processing, it can pass through the following escalation scenario:

◼ As first escalation stage, the system displays a reminder to its operator indicating that processing of the run has been due for a certain amount of time, but not started yet.

◼ If the run still remains unprocessed for some time, it can enter the next stage of the escalation scenario and be declared overdue.

◼ If an overdue run is not processed within a certain time, it eventually expires and is no longer available for processing.

Each stage of the escalation scenario is configurable with respect to when it occurs and if it represents an exception.

System action	Comment
Run due reminder	The system indicates to its operator that a run has been due for some time and ought to be processed.
Run overdue	The system indicates to its operator that a run must be processed immediately.
Run expired	The system removes the run from the list of runs available for processing.

Attribute	Type	Comment
Duration before action (since trigger)	Duration	Available for all actions. Defines the amount of time that passes between the creation of the run and the system action.
Automatic exception enabled	Flag	Available for all actions. Controls if the system automatically tracks the system action as exception that does not allow operator interaction.

Attribute	Type	Comment
Risk assessment	Choice list	Available for all actions. Defines the risk level of the exception. Since there is no operator interaction for the exception, it is not linked to a signature privilege. Available settings: **None**, **Low**, **Low (mandatory comment)**, **Medium**, **Medium (mandatory comment)**, **High**, **High (mandatory comment)**. Default setting: **High**.
Exception text	Text	Available for all actions. Defines the exception description used during exception handling and within the batch record. Maximum length is 2000 characters.

SERVER-RUN

The **Server-run** capability is only available for operations that do not hold any of the other operation capabilities (**Detachable**, **Event-triggered**, **Trigger-enabled**, **Escalation-enabled**, **Auto-startable**). It defines that during execution of an order with PharmaSuite for Production Execution, all phases of the operation run on the system's OE server (Operation Execution server) and not on the client computer on which the other phases of the order are executed. This means that only phases that do not require any user interaction and thus do not have a graphical user interface can be used in an operation with this capability.

> **TIP**
> Please note that for phases without user interface there is no preview available.

The **Server-run** capability is especially suited for operations containing trigger phases, which generate the trigger events that control the runs of **Event-triggered** operations. By having trigger phases run on a server and not on a client you increase their reliability as they do not depend on any operator interaction and are not affected by situations that would usually interrupt processing, such as a user change.

Since a **Server-run** operation and its phases are invisible to operators, the capability does not have attributes to control its behavior. Assigning the capability specifies that the operation runs on the OE server and enables checks to ensure that

- it is the only capability assigned to the operation,
- all of its phases are intended for being run on a server, and
- there is no phase completion privilege parameter assigned to any of its phases.

> **TIP**
>
> You can verify if a phase is suitable for being run on a server by opening its **Source** property window (page 53) and checking in the **System Source** section that the **Server-run** property is selected.

▷ AUTO-STARTABLE

The **Auto-startable** capability is available for operations, but with the following restrictions:

- Operations may not be located on simultaneous branches.

- Operations may not hold the **Event-triggered**, **Escalation-enabled**, or **Server-run** capabilities.

It defines that during execution of an order with PharmaSuite for Production Execution, the operation starts automatically and displays its first phase as soon as an operator has selected its unit procedure for processing.

> **TIP**
>
> Please note that the system ignores the **Auto-startable** setting when the unit procedure that holds the **Auto-startable** operation runs at more than one station or work center, since it cannot determine at which station or work center the operation needs to be started.
> If PharmaSuite for Production Execution is not up and running at the station or work center where the operation is to be processed, the operation has to be started manually once the system is functional.

When you open the Parameter Panel, it lists all parameters available at this level of your recipe or building block. Use the Search (page 13) and Filter (page 14) tools to restrict the number of items shown in the **Results** list.

By default, the parameters are sorted according to the hierarchy of their paths, but you can adjust both the sort and column order (page 14).
Table cells that contain editable values are displayed in black font, while read-only cells are shown in gray.
To set a parameter's values to read-only, select the checkbox in the **Lock** column. The box itself remains editable, so to revoke the setting, uncheck the box.

■ To edit the values of the parameter attributes, proceed as follows:

1. Click the table cell you wish to edit. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

2. Enter the value of the parameter attribute.

3. Press the ENTER key or click in another cell to conclude the edit action.

■ To lock one or more parameters, proceed as follows:

1. Select or multi-select the parameters you wish to lock.

2. Click the **Lock** button (see "Dialog Buttons" in Vol. 1).

■ To unlock one or more parameters, proceed as follows:

> **TIP**
> Please note that you cannot unlock **Frozen** parameters.

1. Select or multi-select the parameters you wish to unlock.

2. Click the **Unlock** button (see "Dialog Buttons" in Vol. 1).

■ To remove one or more parameters, proceed as follows:

> **TIP**
> Please note that you can only remove parameters that are not locked.

1. Select or multi-select the parameters you wish to remove from the list of parameters displayed in the **Results** list.

2. Click the **Remove parameter** button (see "Dialog Buttons" in Vol. 1).

Process Parameters

Process parameters are used to configure recipe or building block elements by specifying their necessary input attributes. Each parameter attribute has its specific data type, such as a string (of characters) for an instruction text. The attribute value you enter, such as "Read the SOP" for an instruction text, is later displayed to an operator during execution. Some phases do not provide only pre-defined process parameters but also allow you to add process parameter bundles. They represent a set of process parameters that supplies a specific behavior to the phase. Specific behavior, for example, can be checking a value of the MeasuredValue data type gathered from execution against limits and expected values.

Instead of specifying a fixed value to be displayed or used during execution, you can also use an expression created in the Expression editor (see "Expression Editor for Process Parameter Attributes" in Vol. 1) to draw the output of another phase or the calculated result of several phase outputs as value into a parameter attribute. When you reference phase outputs in this manner you need to be aware of the following restrictions:

■ Only if a phase has been processed does it provide an output that can be fed into another phase as attribute value. For this reason, you can never reference an output of a phase that is a strict successor of the phase in which you try to use the output.

■ Branches and loops, however, require special notice in this context, since they are only potentially passed through and/or completed during processing, so their outputs are not reliably available. Thus you can reference any such potentially available outputs, but need to be aware of the fact that the provided value may be **Undefined** so that the phase to which you are feeding the output must be able to deal with such an **Undefined** input value.

The basic rule for outputs from operations with internal or external loops is that it is always and only the data from the last or current run of a loop that is considered for feeding into a successor component. Depending on the structure level on which a loop is located, however, the system takes different views as to what constitutes a last run. So when an operation contains an internal loop around a selection branch, it provides the value recorded at the last time the specific path of the branch was passed through. An **Undefined** value can only occur if the path has not yet been passed through.
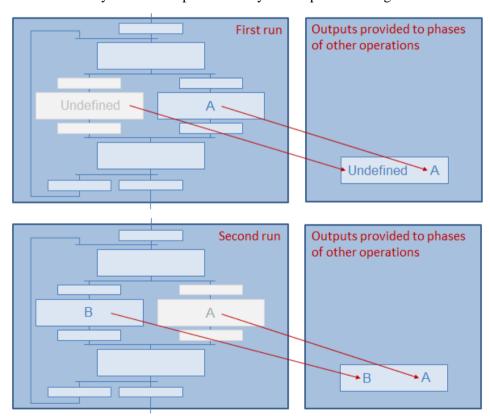
Figure 32: Outputs from operation-internal loops

When an operation contains a selection branch but is also part of an external loop, so that the complete operation can be passed through several times, the system only considers the path or paths that are actually being taken during processing. Thus, only those outputs provide data that have been passed during the last or current run while all other outputs provide **Undefined** as values.
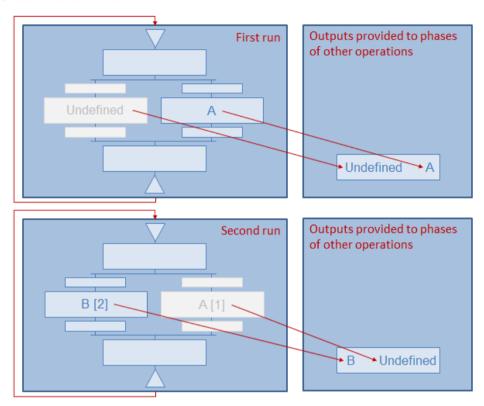
Figure 33: Outputs from operation-external loops

> **TIP**
>
> Please note that event-triggered operations are considered to have implicit external loops. The system uses the count of the individual runs to determine the last or current run. So it is the run with the highest count that provides its output value to successor operations.

When you open the Parameter Panel, it lists all parameters available at this level of your recipe or building block. Use the Search (page 13) and Filter (page 14) tools to restrict the number of items shown in the **Results** list.

In the list, each process parameter is displayed as a nested table, with the common parameter attributes as main table and each parameter's specific attributes in a collapsible sub-table. If a parameter attribute has an expression defined, its cell content is prefixed with an equals sign and the **fx** button (see "Dialog Buttons" in Vol. 1) is permanently displayed for the cell.

TIP

To expand or collapse all process parameter sub-tables, use the keyboard shortcuts CTRL + PLUS (to expand) and CTRL + MINUS (to collapse).

In order to provide a comprehensive overview of a parameter in just one table row, the **Contents** column in the main table contains an abbreviated version of the most important attributes of the parameter.

By default, the parameters are sorted according to the hierarchy of their paths, but you can adjust both the sort and column order (page 14).
Table cells that contain editable values are displayed in black font, while read-only cells are shown in gray.
To set a parameter's values to read-only, select the checkbox in the **Lock** column. The box itself remains editable, so to revoke the setting, uncheck the box.

■ To edit the values of the parameter attributes, proceed as follows:

1. Click the table cell you wish to edit. The system displays a text cursor or a button to access a suitable editor (see "Editors" in Vol. 1) for the required value type.

2. Enter the value of the parameter attribute.

3. Press the ENTER key or click in another cell to conclude the edit action.

TIP

Please note that you can directly edit cells filled from the Measured Value, Multi-line Text, Duration, or DateTime Picker editors (see "Editors" in Vol. 1).
When you type a duration without using the Duration editor, please note the following:

■ Make sure to write the duration as consecutive string of characters without blanks, since otherwise, the system will not be able to interpret it correctly.

■ You can leave out any units you do not need for specifying your duration, so instead of **0d6h0m30s0ms** you can write **6h30s**.

■ You have to observe the order of the units, which means, for instance, that the system cannot interpret a duration that starts with seconds, followed by minutes and hours.

■ The values you specify for the various time units must not exceed the maximum number of each time unit that would move it to the next higher unit. Thus, for hours (**h**) the maximum is 23, for minutes (**min**) and seconds (**s**) it is 59, and for milliseconds (**ms**) it is 999.
This restriction, however, does not apply to the highest unit you specify, which means that the system allows a duration such as **49h30min15s**, whereas typing **90min** in this context would not be valid.
The system automatically converts the highest unit on cell exit, if possible. Thus, it converts **49h30min15s** to **2d1h30min15s**.

■ To edit or delete an expression, however, you have to open the Expression editor (see "Expression Editor for Process Parameter Attributes" in Vol. 1).

■ To lock one or more parameters, proceed as follows:

1. Select or multi-select the parameters you wish to lock.

2. Click the **Lock** button (see "Dialog Buttons" in Vol. 1).

■ To unlock one or more parameters, proceed as follows:

> **TIP**
> Please note that you cannot unlock **Frozen** parameters.

1. Select or multi-select the parameters you wish to unlock.

2. Click the **Unlock** button (see "Dialog Buttons" in Vol. 1).

In addition to the permanent process parameters that are always present, some phases also provide optional process parameter bundles, which you can insert if required. A process parameter bundle is a group of one or more semantically connected process parameters, all of which are needed to produce a specific behavior or function when the phase is executed.

■ To add a process parameter bundle to the list of process parameters, proceed as follows:

1. Click the **Add parameter bundle** button.
   The system opens an option list that holds all bundle types available for the phase.

2. Select the type.
   The system opens the **Add <Bundle Type>** dialog to define the bundle's identifier.

3. Type an identifier and click the **OK** button.
   The system adds all process parameters of the bundle to the list of parameters.

> **TIP**
> Please note that the identifier of the parameter bundle is used as identifier of the bundle's master parameter and as prefix to all other parameters of the bundle.

■ To remove a process parameter bundle from the list of process parameters, proceed as follows:

1. In the list of parameters, select any one of the parameters of the bundle you wish to remove.
   All parameter identifiers of a bundle start with the bundle's identifier.

2. Click the **Remove parameter** button.
   The system asks you to confirm the action and then removes all parameters of the bundle.

> **TIP**
> Please note that you cannot remove individual parameters of a bundle.

Transitions

When transitions (page 151) are specified by at least an identifier they are treated as parameters in that they are listed in the Parameter Panel. Like any other parameter, transitions are listed with their attributes and can assume the three locking states **Unlocked**, **Locked**, and **Frozen**. You can access the Expression editor (see "Expression Editor for Transition conditions" in Vol. 1) for editing the attributes and the transition conditions.

When you open the Parameter Panel, it lists all parameters available at this level of your recipe or building block. Use the Search and Filter tools to restrict the number of items shown in the **Results** list.

By default, the parameters are sorted according to the hierarchy of their paths, but you can adjust both the sort and column order.
Table cells that contain editable values are displayed in black font, while read-only cells are shown in gray.
To set a parameter's values to read-only, select the checkbox in the **Lock** column. The box itself remains editable, so to revoke the setting, uncheck the box.

■ To edit the transition's attributes and its condition, open the Expression editor (see "Expression Editor for Transition conditions" in Vol. 1).

■ To lock one or more transitions, proceed as follows:

1. Select or multi-select the transitions you wish to lock.

2. Click the **Lock** button (see "Dialog Buttons" in Vol. 1).

■ To unlock one or more transitions, proceed as follows:

> **TIP**
> Please note that you cannot unlock **Frozen** transitions.

1. Select or multi-select the transitions you wish to unlock.

2. Click the **Unlock** button (see "Dialog Buttons" in Vol. 1).

Statistics

The **Statistics** function supports you with estimating the size and complexity of your recipe or building block. It provides you with exact counts of its component objects.

> **TIP**
>
> Please note that Recipe Designer is currently not designed for handling recipes that hold more than 1000 phases. If your recipe or building block exceeds this limit, the system displays a corresponding warning.

The Statistics dialog lists the counts of following data objects:

- SFC graph components:
  - Unit procedures
  - Operations
  - Phases
- Parameters
  - Process parameters
  - Material parameters
  - MFC items:
    - Material inputs
    - Material outputs
    - Material transfers
  - Privilege parameters
  - Capability parameters
  - Work center assignment parameters
  - Equipment requirement parameters
  - Additional property requirements
- Expressions defined in the expression editor:
  - Transitions with identifiers
  - Process parameter input expressions
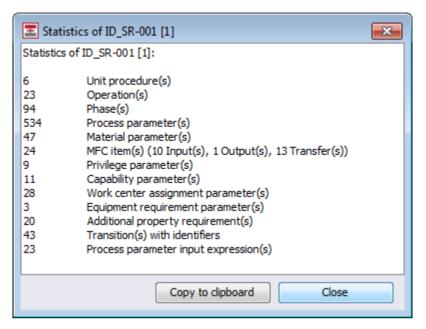
Figure 34: Statistics dialog

The **Statistics** dialog provides the following functions:

- To copy the complete list to the clipboard for subsequent pasting into a spreadsheet application or a text editor, click the **Copy to clipboard** button. To retain its basic formatting in the application into which it is pasted, the list holds blanks and tabs as separator characters.

- To close the dialog, click the **Close** button.

Help Access

The help system of Recipe Designer is context-sensitive on panel and window level and, in the Graph Window, on component level. This means that clicking the help button or pressing the F1 key opens a help window that displays information relevant to the panel or graph component you have currently focused. The help window is non-modal and resizable.
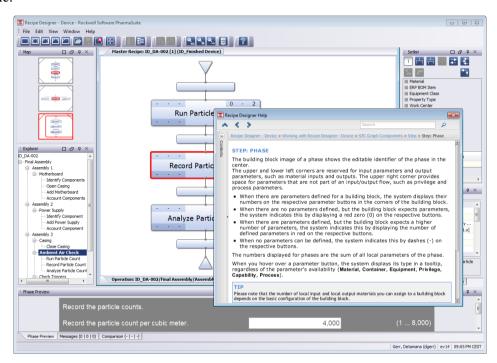
Figure 35: Recipe Designer with help window

In order to access other topics than the one directly related to the current context, use the navigation arrows located at the top of the help page, or related topic links located at the bottom of the page, if available.

The following additional features support your use of the help system:

■ To access an overview of all available topics open the **Contents** tab in the **Contents and Index** frame. The system additionally provides **Expand all** and **Collapse all** buttons to facilitate easier navigation in the contents tree.

■ To access the index, open the **Index** tab in the **Contents and Index** frame.

■ To use the **Search** function

1. type the term you are looking for in the **Search** box and

2. click the **Search** button or press the ENTER key.
   The system will display all occurrences of the search term in a third tab in the **Content and Index** frame.

■ To print the page that is currently displayed in the help window, click the **Print this page** button. The system displays a print preview of the page along with the default Windows **Print** dialog.

> **TIP**
> Please note that printing is only available from the stand-alone format of the help system and not from within the application.
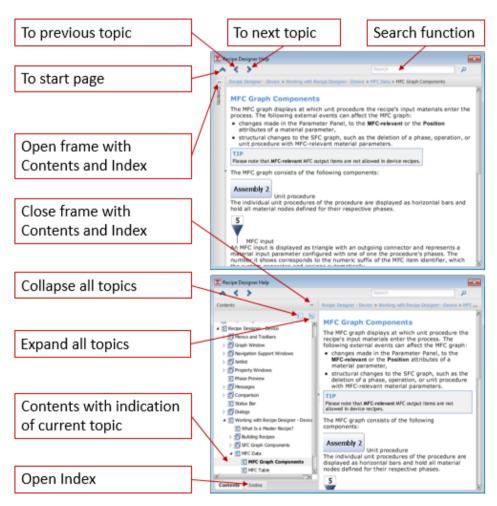
Figure 36: Help window navigation

About PharmaSuite

The **About PharmaSuite** function opens the **About PharmaSuite** dialog to display system-related information, such as the current system version and build, the logged-in user, work center, and database-related information.

> **TIP**
>
> Please note that the dialog also indicates the **EBR server state**. Only when the EBR server is available can orders or workflows be processed in PharmaSuite for Production Execution.

Click the **Details** button to view more specific technical information on the system and its environment.
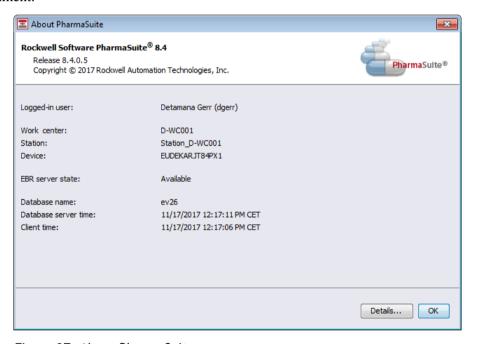
Figure 37: About PharmaSuite

Details

From the **PharmaSuite Installation Details** dialog, you can copy the listed detail data to the clipboard.

> **TIP**
>
> Please note that the path to the PharmaSuite log files is given in the last section of the listed detail data.
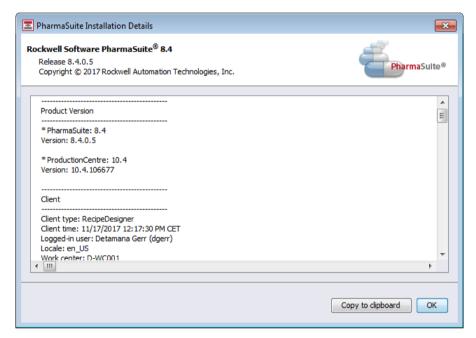
Figure 38: PharmaSuite Installation Details

Working with Recipe Designer - Device

Within in the framework of Recipe and Workflow Designer that is based on the concepts of the S88 and S95 standards, Recipe Designer provides the functions for creating and maintaining master recipes (page 131) with their component building blocks, including the flow of materials (see "What Is Material Flow Control (MFC)?" in Vol. 1) through the unit procedures of a master recipe or procedure building block.

What Is a Master Recipe?

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.

A master recipe for a device order must be fully configured with all parameters, such as process parameters, equipment requirement parameters, and material inputs. Only then is it available in PharmaSuite for Production Management for creating an order, which needs to be exploded and released before it can be selected for processing on the shop floor.

Building Recipes

Before you can start to work on a master recipe you have to open an existing master recipe if you wish to modify it or create a new master recipe from scratch.

To open an existing master recipe, proceed as follows:

1. In the main menu bar (page 17), open the **File** menu and select **Open** or in the File toolbar (page 26), click the **Open** button.
   The system displays the **Open** dialog (page 83).
   In the **Filter** tool of the **Open** dialog, the object type is preset to **Master Recipe**.

> **TIP**
> Please note that your access rights determine which master recipes are available for selection. Confidential master recipes that you are not allowed to access are not shown in the list.

2. Double-click the master recipe you wish to work on.
The system opens a new tab in the upper tab bar of the Graph Window (page 29).
It represents the master recipe level (page 138), which is the highest level of the master recipe structure, and displays the procedure of the master recipe (page 143).

> **TIP**
>
> Whether or not you can edit a master recipe depends on its status. Only master recipes in the **Edit** status can be edited. Editable master recipes with unsaved changes show an asterisk marker (*) on their upper tab title, while read-only master recipes are marked by an (R).

To create a new master recipe, proceed as follows:

1. In the main menu bar (page 17), open the **File** menu and select **New master recipe** or in the File toolbar (page 26), click the **New master recipe** button.
The system displays the **Select Material** dialog (page 84).

2. You can now decide to build a new, independent master recipe or to base it on an existing ERP BOM.
In the **Object type** column of the Filter tool,

   ■ select **Material** if you wish to create an independent master recipe.

   ■ select **Material (ERP BOM)** to base the recipe on an existing ERP BOM (page 137).

> **TIPS**
>
> Please note that the system only lists materials that are measured either in **ea** or a unit that is convertible into **ea**.
> Please note that your access rights determine which materials are available for selection. Product materials for confidential recipes that you are not allowed to access are not shown in the list.

Double-click the material you wish to produce with your new master recipe.
The system closes the **Select Material** dialog and displays the **New Master Recipe** dialog.

3. The **Identifier** is preset with the identifier of the material that will be produced. You can suffix, prefix, or edit the preset identifier. Type the **Identifier** of the new master recipe in the box.
If the selected material has an access privilege for confidential objects assigned or the selected ERP BOM is protected by such an access privilege, the access privilege is passed on to the master recipe and cannot be changed.
If the master recipe is confidential and needs to have restricted access permissions, you can select a suitable access privilege, even if the selected material does not have an access privilege for confidential objects assigned or the selected ERP BOM is not protected by such an access privilege. The list of

available access privileges only contains those privileges that are also assigned to your current login.

4. Click the **OK** button.
    The system opens a new tab in the upper tab bar of the Graph Window (page 29). It represents the master recipe level (page 138), which is the highest level of the master recipe structure, and displays a start (page 142) and an end (page 142) step enclosing a procedure building block (page 143), which you can configure for your new recipe.

5. Double-click the procedure building block to create the procedure level (page 138) with a unit procedure building block (page 144). The Graph Window updates to display the new building block enclosed by a start (page 142) and end (page 142) step.

6. Double-click the unit procedure building block to create the unit procedure level (page 140) with an operation building block (page 146). The Graph Window updates to display the new building block enclosed by a start (page 142) and end (page 142) step.

7. Double-click the operation building block to create the operation level (page 141) with a phase building block (page 149).The Graph Window updates to display the new building block enclosed by a start (page 142) and end (page 142) step.
    In the Map (page 33) and the Explorer (page 34), you can watch the master recipe structure grow.

To copy an existing master recipe, proceed as follows:

1. Open the master recipe you wish to copy.

2. In the main menu bar (page 17), open the **File** menu and select **Save <master recipe identifier> as**.
    The system displays the **Save as** dialog.

3. The **Identifier** box is pre-filled with the identifier of the master recipe you are about to copy. Adjust the identifier or type a new one.
    If the master recipe is confidential and needs to have restricted access permissions, it inherits its access privilege from the original master recipe. The inherited access privilege cannot be changed. If the original master recipe is not confidential, you can select a suitable access privilege. The list of available access privileges only contains those privileges that are also assigned to your current login.

4. Click the **OK** button.
    The system updates the tab title in the upper tab bar of the Graph Window and displays the new master recipe in the **Edit** status.

To rename an existing master recipe, proceed as follows:

1. Open the master recipe you wish to rename.

2. In the main menu bar (page 17), open the **File** menu and select **Rename <master recipe identifier>**.
   The system displays the **Rename Master Recipe** dialog.
   The **Identifier** is preset with the current identifier of the master recipe.

3. Adapt the identifier or type a new one, then click the **OK** button.

To replace the material to be produced with a master recipe, proceed as follows:

1. Open the master recipe whose product material you wish to change.

2. In the main menu bar (page 17), open the **File** menu and select **Replace material (product) of <master recipe identifier>**.
   The system displays a **Question** dialog asking you to confirm that you wish to proceed with the function you are about to perform.

3. Click **Yes** to proceed.
   The system displays the **Select Material** dialog where you can decide to either select just a material or a material that is connected to an ERP BOM (page 137).

   > **TIP**
   > Please note that your access rights determine which materials are available for selection. Product materials for confidential recipes that you are not allowed to access are not shown in the list.

4. Double-click the material you wish to produce from now on with your master recipe.
   The system closes the **Select Material** dialog.
   If the selected material has an access privilege for confidential objects assigned or the selected ERP BOM is protected by an access privilege for confidential objects that differs from the master recipe's current one, the master recipe is updated and from then on protected by the new access privilege.
   Changing the product material of a master recipe affects its MFC data. All issues that may result from the change are listed in the Messages window (page 57).

When you have created a new master recipe structure from scratch, your Graph Window (page 29) holds the basic graph structure with one dummy building block on each level of the hierarchy. To build your master recipe, proceed as follows:

1. In the View toolbar (page 26), click the **Open Universe** button (see "Toolbar Buttons" in Vol. 1).
   The system opens the Universe (page 94) where you can select all the components, inputs, and outputs you wish to use in your master recipe and feed them into the Setlist (page 35).

2. Navigate (page 33) to the structure level (page 138) on which you wish to build the graph.

3. Use the toggle buttons of the Setlist toolbar (page 27) to define how you will add a Setlist item to the graph (sequence, selection branch (page 155), simultaneous branch (page 156)).

   > **TIP**
   > Please note that to replace the first, system-generated element of a structure level with a building block from the Setlist you can use the **Replace** button (see "Toolbar Buttons" in Vol. 1).

4. Double-click a building block in the Setlist to add it in the defined manner to your graph.

   > **TIPS**
   > Please note that only in Graph Window can you insert loops.
   > Right-click the building block where you wish the loop to start, to open the process workflow toolbar (page 27). Click the **Loop** button (see "Toolbar Buttons" in Vol. 1) and select a suitable loop endpoint to close the loop.

5. If your graph contains branches or loops, configure the transition conditions (page 151) that are relevant to the decision points.

6. Use the toggle buttons of the Setlist toolbar (page 27) and double-click the respective parameter to add local input and output materials (page 95), equipment requirement parameters (page 101), work center assignment parameters (page 106), privilege parameters (page 107), capabilities (page 109), or process parameters (page 118) to your elements.

7. To configure the parameters of the elements, click one of their parameter buttons located in the corners of the building block.
   The system opens the Parameter Panel (page 95) where you can define all parameter properties.

   > **TIP**
   > If your recipe contains material parameters that need to be tracked by material flow control, be sure to mark them as **MFC-relevant** by selecting this option in the Parameter Panel.

8. In the main menu bar (page 17), open the **View** menu and select the **Material Flow Control** function or use the keyboard shortcut ALT+C to open the **Material Flow Control** tab (page 158) in the lower tab bar. Now define how the materials you have marked as MFC-relevant flow through the unit procedures of your procedure graph.

9. When you have completed all graphs of your recipe and have configured all parameters, check if the master recipe's structure, parameters, and MFC definitions are valid.
   Open the Messages window (page 57) to view all messages the system has returned for your master recipe. Make all necessary changes.

   > **TIP**
   > To clean up the alignment and spacing of steps within your SFC graphs, use the **Arrange** and **Compact** functions available from the shortcut menus on each structure level (page 24) of your master recipe.

10. When your master recipe has passed all checks successfully and the Messages window does not list any messages, it is ready to be prepared for the next steps in its life-cycle, such as verification.
    In the main menu bar (page 17), open the **File** menu and select **Prepare <master recipe identifier> for status change**.

    > **TIP**
    > Depending on the size and complexity of your master recipe, this function may take some time to complete. Only when it has been prepared, can you perform status changes on the master recipe.

11. Your master recipe now is ready to be handed over for verification. For this purpose you want it to be read-only to make sure no changes can be applied to the master recipe any longer. To achieve this, you need to change its status (page 88).
    In the main menu bar (page 17), open the **File** menu and select **Change status of <master recipe identifier>**.

12. Select the **Action** that moves your master recipe to the **Verification** status (page 87).
    Once the recipe has been verified successfully, it can be moved to the **Valid** status and thus made available for use in order creation with PharmaSuite for Production Management and eventually order processing with PharmaSuite for Production Execution.

Working with ERP BOMs

In many cases, recipes are not created from scratch in Recipe Designer and maintained only in PharmaSuite, but are based on BOMs that are stored in a superordinate ERP system. ERP BOMs can be imported into PharmaSuite and made available for your reference when you create a new recipe in Recipe Designer.

To base a new master recipe on an existing ERP BOM, proceed as follows:

1. From the File menu, select **New master recipe** or click the **New master recipe** toolbar button to open the **Select Material** dialog.

2. In the **Object type** column of the Filter tool, select **Material (ERP BOM)**. The system lists all materials for which an ERP BOM is available.

3. Double-click to select the material to be produced. The selection assigns the ERP BOM to your new master recipe.
   In the Setlist (page 35), the system lists all items of the assigned ERP BOM under the **ERP BOM Item** node, sorted by their BOM positions. The system expects the items of the ERP BOM as MFC inputs of the master recipe. Additionally, the system checks if the positions and quantities of the material input parameters correspond to the values defined in the assigned ERP BOM. Mismatch errors are shown as entries in the Messages window (page 64). So, initially, all ERP BOM items are indicated as missing.

   > **TIP**
   > To resolve each of these messages individually, right-click it to open the messages shortcut menu and select the **Resolve issue** function.
   > To resolve all of the issues in one go, select the **Resolve all issues** function of any of the related messages.

4. Use the Setlist to add the material parameters to the appropriate phases of your master recipe. When you double-click an ERP BOM item to add it as material parameter, the system also populates the **BOM position** and **Planned quantity** attributes as defined in the ERP BOM item. Make sure the **MFC-relevant** option is also selected.

5. To view data relevant to the ERP BOM, such as the identifier and short description of the produced material and the key data of the ERP BOM (**Alternative**, **Base quantity**, and **Release** state) as maintained in the ERP system, refer to the **Material (Product)** section displayed at the bottom of the master recipe's **Header** property window (page 45).

> **TIP**
> To change the ERP BOM of an existing recipe, use the **Replace material (product)** function (page 17) from the **File** menu.

SFC Graph Components

Recipe Designer provides all components and actions required for building SFC graphs on all levels of a recipe or building block hierarchy.

Structure Level

The hierarchical structure of a recipe is reflected in Recipe Designer by separate tabs for each level.

STRUCTURE LEVEL: MASTER RECIPE

A master recipe (page 131) tab contains exactly one procedure step that is enclosed by a start and an end step.
To rename the master recipe, open the **File** menu and select the **Rename <master recipe identifier>** function.

Right-click anywhere on the background to open a shortcut menu with the following functions:

> **TIP**
>
> Which of the functions are enabled also depends on the graph components you have selected.

- Rename selected (F2)
  In an SFC graph tab, makes the identifier of the currently selected component available for editing. Press the ESC key to cancel this action.

- Create building block from selected (F12)
  In an SFC graph tab, opens the **Create <Object Type>** dialog where you can save your parameterized building block to make it available for selection from the Universe.

- Go back (ALT+LEFT)
  Returns to the previously active tab in the lower tab bar within the currently active main component, if you navigated to your current tab from another one.

- Close sub-tab (CTRL+W)
  Closes the currently active tab of the lower tab bar.

STRUCTURE LEVEL: PROCEDURE

A procedure (see "What Is a Procedure?" in Vol. 1) tab contains a graph that consists of unit procedures, their transitions, and links, enclosed by a start and an end step.

TIPS

On this level, loops are not permitted and will fail the validity check of the master recipe.

Please be aware that structural changes to a graph, such as deleting components that hold MFC-relevant material parameters or swapping places between such components, affects the material flow defined in the MFC tab (page 158). Issues that may arise from changes of this kind are listed in the Messages window (page 57).

Right-click anywhere on the background to open a shortcut menu with the following functions:

TIP

Which of the functions are enabled also depends on the graph components you have selected.

■ New step (unconnected) (INS)
Inserts a new unconnected dummy unit procedure at the current cursor position.

■ Rename selected (F2)
In an SFC graph tab, makes the identifier of the currently selected component available for editing. Press the ESC key to cancel this action.

■ Create building block from selected (F12)
In an SFC graph tab, opens the **Create <Object Type>** dialog where you can save your parameterized building block to make it available for selection from the Universe.

■ Delete selected (DEL)
In an SFC graph tab, deletes all currently selected components.

■ Arrange
Arranges the graph and its components to center sequences and branches with respect to their start components and to optimize the vertical spacing between individual components.

■ Compact graph
Compacts the graph for optimal use of space.

■ Go back (ALT+LEFT)
Returns to the previously active tab in the lower tab bar within the currently active main component, if you navigated to your current tab from another one.

■ Go up (ALT+UP)
Moves to the next higher structure level.

■ Close sub-tab (CTRL+W)
Closes the currently active tab of the lower tab bar.

STRUCTURE LEVEL: UNIT PROCEDURE

A unit procedure (see "What Is a Unit Procedure?" in Vol. 1) tab contains a graph that consists of operations, their transitions, and links, enclosed by a start and an end step.

> **TIP**
>
> Please be aware that structural changes to a graph, such as deleting components that hold MFC-relevant material parameters or swapping places between such components, affects the material flow defined in the MFC tab (page 158). Issues that may arise from changes of this kind are listed in the Messages window (page 57).

Right-click anywhere on the background to open a shortcut menu with the following functions:

> **TIP**
>
> Which of the functions are enabled also depends on the graph components you have selected.

- **New step (unconnected)** (INS)
  Inserts a new unconnected dummy operation at the current cursor position.

- **Rename selected** (F2)
  In an SFC graph tab, makes the identifier of the currently selected component available for editing. Press the ESC key to cancel this action.

- **Create building block from selected** (F12)
  In an SFC graph tab, opens the **Create <Object Type>** dialog where you can save your parameterized building block to make it available for selection from the Universe.

- **Delete selected** (DEL)
  In an SFC graph tab, deletes all currently selected components.

- **Arrange**
  Arranges the graph and its components to center sequences and branches with respect to their start components and to optimize the vertical spacing between individual components.

- **Compact graph**
  Compacts the graph for optimal use of space.

- **Go back** (ALT+LEFT)
  Returns to the previously active tab in the lower tab bar within the currently active main component, if you navigated to your current tab from another one.

- **Go up** (ALT+UP)
  Moves to the next higher structure level.

■ Close sub-tab (CTRL+W)
Closes the currently active tab of the lower tab bar.

STRUCTURE LEVEL: OPERATION

An operation (see "What Is an Operation?" in Vol. 1) tab contains a graph that consists of phases, their transitions, and links, enclosed by a start and an end step.

> **TIP**
>
> Please be aware that structural changes to a graph, such as deleting components that hold MFC-relevant material parameters or swapping places between such components, affects the material flow defined in the MFC tab (page 158). Issues that may arise from changes of this kind are listed in the Messages window (page 57).

Right-click anywhere on the background to open a shortcut menu with the following functions:

> **TIP**
>
> Which of the functions are enabled also depends on the graph components you have selected.

■ New step (unconnected) (INS)
Inserts a new unconnected dummy phase at the current cursor position.

■ Rename selected (F2)
In an SFC graph tab, makes the identifier of the currently selected component available for editing. Press the ESC key to cancel this action.

■ Create building block from selected (F12)
In an SFC graph tab, opens the **Create <Object Type>** dialog where you can save your parameterized building block to make it available for selection from the Universe.

■ Delete selected (DEL)
In an SFC graph tab, deletes all currently selected components.

■ Arrange
Arranges the graph and its components to center sequences and branches with respect to their start components and to optimize the vertical spacing between individual components.

■ Compact graph
Compacts the graph for optimal use of space.

■ Go back (ALT+LEFT)
Returns to the previously active tab in the lower tab bar within the currently active main component, if you navigated to your current tab from another one.

■ Go up (ALT+UP)
Moves to the next higher structure level.

■ Close sub-tab (CTRL+W)
Closes the currently active tab of the lower tab bar.

Step

Depending on their level within the graph structure, graph steps need to be handled differently and can provide various additional functions.

STEP: START

A start step represents the first component of a graph on each structure level. It can be moved and selected for inserting components below it, but it cannot be deleted.
The start step allows you to begin a graph with parallel building blocks and to create a loop that includes the first building block of your graph.

When you hover over a start step, the system displays its meaning in a tooltip.
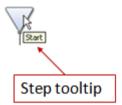
Figure 39: Start step

STEP: END

An end step represents the final component of a graph on each structure level. It can be moved and selected for inserting components above it, but it cannot be deleted.
The end step allows you to create a loop that includes the final building block of your graph.

When you hover over an end step, the system displays its meaning in a tooltip.
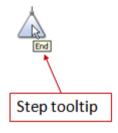
Figure 40: End step

STEP: PROCEDURE

The element image of a procedure shows the editable identifier of the procedure in the center.

The upper and lower left corners are reserved for input parameters and output parameters, such as material inputs and outputs. The upper right corner provides space for parameters that are not part of an input/output flow, such as privilege and process parameters.

- When there are parameters defined for a building block, the system displays their numbers on the respective parameter buttons in the corners of the building block.

- When there are no parameters defined, but the building block expects parameters, the system indicates this by displaying a red zero (0) on the respective buttons.

- When there are parameters defined, but the building block expects a higher number of parameters, the system indicates this by displaying the number of defined parameters in red on the respective buttons.

- When no parameters can be defined, the system indicates this by dashes (-) on the respective buttons.

The numbers displayed for procedures are the sum of all local parameters of the procedure plus all the parameters of its child elements (unit procedures, operations, and phases).

The count of equipment parameters sums up all equipment classes, equipment entities, and individual property types assigned to the phases of the procedure as well as all work centers assigned to its unit procedures.

When you hover over a parameter button, the system displays its type in a tooltip, regardless of the parameter's availability (**Material**, **Container**, **Equipment**, **Transition**, **Privilege**, **Capability**, **Process**).
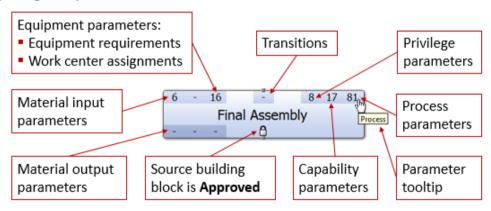
Figure 41: Procedure element

> **TIP**
>
> Since a master recipe can only have one procedure, you cannot add further procedure steps or draw any connectors at this level.

The system provides the following functions for procedure (see "What Is a Procedure?" in Vol. 1) steps:

- To open the Parameter Panel (page 95) for configuring the parameters of the procedure, left-click a parameter button, indicated by the **Select parameter** cursor (see "Cursors" in Vol. 1).
  The number displayed on the parameter buttons indicates how many parameters of this class are assigned to the procedure and all of its subordinate unit procedures, operations, and phases.

  > **TIP**
  > Please note that work center and station assignments are given as part of the count of equipment parameters.

- To open the procedure graph and display its unit procedures in the same tab, double-click the procedure.

- To open the procedure graph and display its unit procedures in a new tab, CTRL+double-click the procedure.

- To edit the identifier of the procedure:

  - from the **Edit** menu, select **Rename selected**, or

  - from the SFC Graph Window, access the shortcut menu, and select **Rename selected**, or

  - slowly click the identifier twice, or

  - press the F2 button.

  > **TIP**
  > Please note that curly brackets (braces) and leading or trailing blanks are not allowed in procedure identifiers.

STEP: UNIT PROCEDURE

The building block image of a unit procedure shows the editable identifier of the unit procedure in the center.
The upper and lower left corners are reserved for input parameters and output parameters, such as material inputs and outputs. The upper right corner provides space for parameters that are not part of an input/output flow, such as privilege and process parameters.

- When there are parameters defined for a building block, the system displays their numbers on the respective parameter buttons in the corners of the building block.

■ When there are no parameters defined, but the building block expects parameters, the system indicates this by displaying a red zero (0) on the respective buttons.

■ When there are parameters defined, but the building block expects a higher number of parameters, the system indicates this by displaying the number of defined parameters in red on the respective buttons.

■ When no parameters can be defined, the system indicates this by dashes (-) on the respective buttons.

The numbers displayed for unit procedures are the sum of all local parameters of the unit procedure plus all the parameters of its child elements (operations and phases).
The count of equipment parameters sums up

■ the work centers assigned to the unit procedure,

■ the stations assigned to the operations of the unit procedure, and

■ the equipment classes assigned as requirements to the phases of the unit procedure.

When you hover over a parameter button, the system displays its type in a tooltip, regardless of the parameter's availability (**Material**, **Container**, **Equipment**, **Transition**, **Privilege**, **Capability**, **Process**).
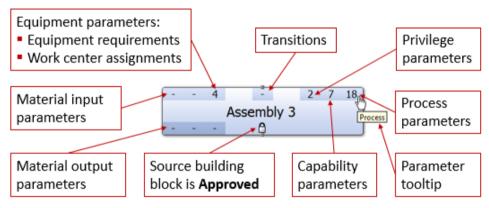
Figure 42: Unit procedure element
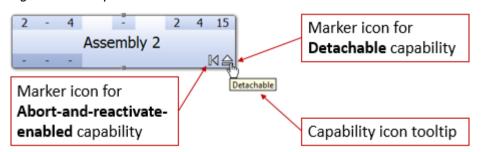
Figure 43: Unit procedure element with capabilities

The system provides the following functions for unit procedure (see "What Is a Unit Procedure?" in Vol. 1) steps:

- To open the Parameter Panel (page 95) for configuring the parameters of the unit procedure, left-click a parameter button, indicated by the **Select parameter** cursor (see "Cursors" in Vol. 1).
  The number displayed on the parameter buttons indicates how many parameters of this class are assigned to the unit procedure and all of its subordinate operations and phases.

  > **TIP**
  > Please note that work center and station assignments are given as part of the count of equipment parameters.

- To build the unit procedure graph, right-click the unit procedure to open the process workflow toolbar (page 27).

- To open the unit procedure graph and display its operations in the same tab, double-click the unit procedure.

- To open the unit procedure graph and display its operations in a new tab, CTRL+double-click the unit procedure.

- To edit the identifier of the unit procedure:

  - from the **Edit** menu, select **Rename selected**, or

  - from the SFC Graph Window, access the shortcut menu, and select **Rename selected**, or

  - slowly click the identifier twice, or

  - press the F2 button.

  > **TIP**
  > Please note that curly brackets (braces) and leading or trailing blanks are not allowed in unit procedure identifiers.

STEP: OPERATION

The element image of an operation shows the editable identifier of the operation in the center.
The upper and lower left corners are reserved for input parameters and output parameters, such as material inputs and outputs. The upper right corner provides space for parameters that are not part of an input/output flow, such as privilege and process parameters. If there are capabilities assigned to the operation, the element image may reflect that by displaying an indicator icon in the lower right corner or changing its image background to a different color or pattern. Most capabilities, however, do not have a specific graphical marker.

- When there are parameters defined for a building block, the system displays their numbers on the respective parameter buttons in the corners of the building block.

■ When there are no parameters defined, but the building block expects parameters, the system indicates this by displaying a red zero (0) on the respective buttons.

■ When there are parameters defined, but the building block expects a higher number of parameters, the system indicates this by displaying the number of defined parameters in red on the respective buttons.

■ When no parameters can be defined, the system indicates this by dashes (-) on the respective buttons.

The numbers displayed for operations are the sum of all local parameters of the operation plus all the parameters of its child phases.
The count of equipment parameters sums up

■ the stations assigned to the operations of the unit procedure and

■ the equipment classes assigned as requirements to the phases of the unit procedure.

When you hover over a parameter button, the system displays its type in a tooltip, regardless of the parameter's availability (**Material**, **Container**, **Equipment**, **Transition**, **Privilege**, **Capability**, **Process**).
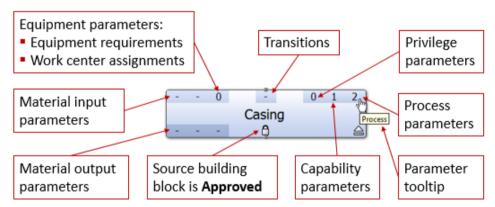
Figure 44: Operation element
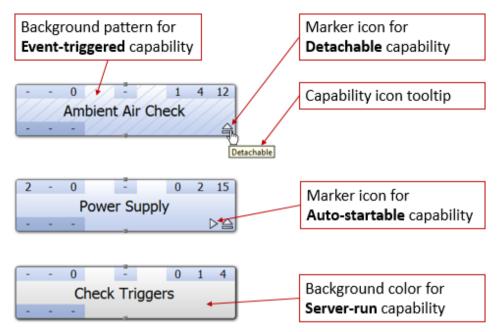
Figure 45: Operation element with capabilities

> **TIP**
>
> Please note that, during the execution of the operation's phases on the shop floor, the system will display the operation identifier as title in the header bar of the Execution Window.

The system provides the following functions for operation (see "What Is an Operation?" in Vol. 1) steps:

■ To open the Parameter Panel (page 95) for configuring the parameters of the operation, left-click a parameter button, indicated by the **Select parameter** cursor (see "Cursors" in Vol. 1).
The number displayed on the parameter buttons indicates how many parameters of this class are assigned to the operation and all of its subordinate phases.

■ To open the Parameter Panel for capabilities (page 109) for configuring the parameters of the operation, left-click a capability marker icon, indicated by the **Select parameter** cursor (see "Cursors" in Vol. 1).
In the list of capabilities, the system highlights the capability whose icon you have clicked.

■ To build the operation graph, right-click the operation to open the process workflow toolbar (page 27).

■ To open the operation graph and display its phases in the same tab, double-click the operation.

■ To open the operation graph and display its phases in a new tab, CTRL+double-click the operation.

■ To edit the identifier of the operation:

   ■ from the **Edit** menu, select **Rename selected**, or

   ■ from the SFC Graph Window, access the shortcut menu, and select **Rename selected**, or

   ■ slowly click the identifier twice, or

   ■ press the F2 button.

> **TIP**
> Please note that curly brackets (braces) and leading or trailing blanks are not allowed in operation identifiers.

STEP: PHASE

The building block image of a phase shows the editable identifier of the phase in the center.

The upper and lower left corners are reserved for input parameters and output parameters, such as material inputs and outputs. The upper right corner provides space for parameters that are not part of an input/output flow, such as privilege and process parameters.

■ When there are parameters defined for a building block, the system displays their numbers on the respective parameter buttons in the corners of the building block.

■ When there are no parameters defined, but the building block expects parameters, the system indicates this by displaying a red zero (0) on the respective buttons.

■ When there are parameters defined, but the building block expects a higher number of parameters, the system indicates this by displaying the number of defined parameters in red on the respective buttons.

■ When no parameters can be defined, the system indicates this by dashes (-) on the respective buttons.

The numbers displayed for phases are the sum of all local parameters of the phase.

When you hover over a parameter button, the system displays its type in a tooltip, regardless of the parameter's availability (**Material**, **Container**, **Equipment**, **Transition**, **Privilege**, **Capability**, **Process**).

> **TIP**
> Please note that the number of local input and local output materials you can assign to a building block depends on the basic configuration of the building block.
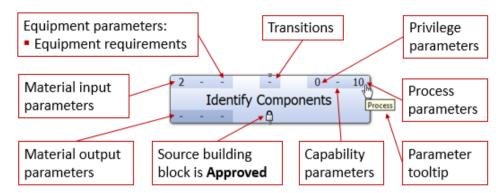
Figure 46: Phase element

The system provides the following functions for phase (see "What Is a Phase?" in Vol. 1) steps:

- To open the Parameter Panel (page 95) for configuring the parameters of the phase, left-click a parameter button, indicated by the **Select parameter** cursor (see "Cursors" in Vol. 1).
  The number displayed on the parameter buttons indicates how many parameters of this class are assigned to the phase.

- To build the phase graph, right-click the phase to open the process workflow toolbar (page 27).

- To edit the identifier of the phase:

  - from the **Edit** menu, select **Rename selected**, or

  - from the SFC Graph Window, access the shortcut menu, and select **Rename selected**, or

  - slowly click the identifier twice, or

  - press the F2 button.

> **TIP**
> Please note that curly brackets (braces) and leading or trailing blanks are not allowed in phase identifiers.

Transition

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.

Generally, all transitions are implicitly preset with a default condition, which simply defines that the step or steps preceding the transition must have been finished before the step or steps after the transition can be started.
A selection branch or loop, however, must be preceded by a specific transition condition so that the system can determine the appropriate processing sequence of steps during order execution.

When a transition holds a specific condition it displays an **fx** marker icon on its transition image. In addition, it requires you to set a transition identifier, either by using the system-generated suggestion or by defining another one. The maximum number of characters for a transition identifier is 20 and it must be unique within its graph. Transition identifiers are only mandatory for transitions with specific conditions or descriptions. For transitions that only hold the default condition, they are optional.

Thus the component image of a transition can either display as blank square, show just an editable transition identifier, or show the transition condition marker and an identifier.
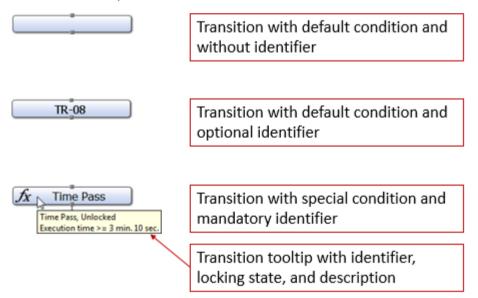
Figure 47: Transition image

TIP

Please note that once a transition has an identifier it is treated as parameter and is thus listed in the Parameter Panel for transitions (page 124).

The following functions are available for transitions:

- To edit the identifier of the transition:

    - from the **Edit** menu, select **Rename selected**, or

    - from the SFC Graph Window, access the shortcut menu, and select **Rename selected**, or

    - slowly click the identifier twice, or

    - press the F2 button.

- To display a tooltip with the transition's identifier, its locking state, and its description, hover over the transition's image in the SFC graph. If there is no description defined, the tooltip shows the condition itself.

- To use the Expression editor (see "Expression Editor for Transition conditions" in Vol. 1) for defining a specific transition condition, double-click the transition. Defining a specific transition condition does not replace the default condition, which always applies to any transition and is therefore not visible in the Expression editor.

TIP

Please note that the transition conditions of a selection branch or loop must be mutually exclusive and collectively exhaustive. This means that for each path after a decision point only one and exactly one of the defined conditions must be applicable each time the process reaches the decision point.
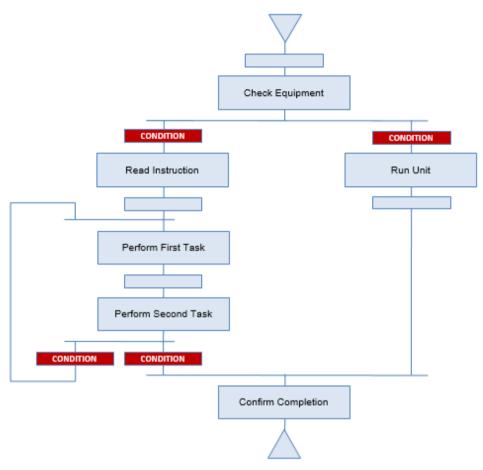
Figure 48: Sample graph with required transition conditions

Link

A link is a simple connector between two components.

Usually, the system draws all necessary links when you use the Setlist or the process workflow toolbar to build your graph. Of course, it is always possible to create or delete a link.

To draw a new link, proceed as follows:

1. Find a suitable port to start the link. The system displays the **draw** cursor (see "Cursors" in Vol. 1) when you hover over a suitable port.

2. Click the port and drag to another suitable port. The draw cursor turns into the **draw endpoint** cursor (see "Cursors" in Vol. 1) when you hover over an allowed endpoint and into the **draw endpoint unallowed** cursor (see "Cursors" in Vol. 1) when the port is not a suitable link endpoint.

3. Drop the link endpoint on the port.
   If necessary the system will add missing components, such as transitions or branches, to maintain the validity of your graph.

> **TIP**
>
> Please note that you can draw links in both directions, with or against the flow of the graph. Regardless of your drawing direction, the system will assume the default flow direction, which is from top to down.
> Explicit upward links, however, as you would draw them for a loop (page 154), between an outgoing port of one step and the ingoing port of another step, which is located further up in the graph, are recognized by the system.

To delete a link, proceed as follows:

1. Select the link you wish to delete.

2. Press the DELETE key or
   right-click on the Graph Window background to select the **Delete selected** function from the shortcut menu.

> **TIP**
>
> You can adjust the layout of a link by moving its corner handles.
> To reset a link to its default layout, right-click the link and select the **Revert to default layout** function from the shortcut menu.

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.

To draw a loop, proceed as follows:

1. Right-click the step from which the loop will start to open the process workflow toolbar (page 27).

2. From the process workflow toolbar, select the **Loop** button. The system displays the **draw** cursor (see "Cursors" in Vol. 1) to indicate that you are in the drawing mode.
   To leave the drawing mode, press the ESC key or right-click anywhere on the background without selecting a function from the shortcut menu.

3. To close the loop, select its end point by performing one of the following actions:

  ◻ In the Graph Window, click the endpoint step. The draw cursor turns into the **draw endpoint** cursor (see "Cursors" in Vol. 1) when you hover over an allowed endpoint step and into the **draw endpoint unallowed** cursor (see "Cursors" in Vol. 1) when the step is not a suitable endpoint.

  ◻ In the Graph Window, press CTRL+F to open the **Search** function and use it to navigate to the endpoint step.

  ◻ In the Explorer window, click the node of the endpoint step.

To delete a loop proceed as follows:

1. Select the loop you wish to delete.

2. Press the DELETE key or
right-click on the Graph Window background to select the **Delete selected** function from the shortcut menu.

TIP

You can adjust the layout of a loop by moving its corner handles.
To reset a loop to its default layout, right-click the loop and select the **Revert to default layout** function from the shortcut menu.

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.

Usually, the system draws all necessary branches when you use the process workflow toolbar to build your graph or when you manually draw a link (page 153) that requires a branch to form a valid graph.

To delete a branch, proceed as follows:

1. Select the branch you wish to delete.

2. Press the DELETE key or
right-click on the Graph Window background to select the **Delete selected** function from the shortcut menu.

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.

Usually, the system draws all necessary branches when you use the process workflow toolbar to build your graph or when you manually draw a link (page 153) that requires a branch to form a valid graph.

To delete a branch, proceed as follows:

1. Select the branch you wish to delete.

2. Press the DELETE key or
right-click on the Graph Window background to select the **Delete selected** function from the shortcut menu.

Special Components

The system provides the following pre-defined components that support you with building the structure of your master recipes and building blocks:

- Dummy Operation (page 156)

- Dummy Phase (page 156)

- Dummy Procedure (page 156)

- Dummy Unit Procedure (page 157)

- Hidden Phase (page 157)

DUMMY OPERATION

A **Dummy Operation** is an operation placeholder that you can pull from the Setlist into your graph as empty element when you build a graph on unit procedure level.

DUMMY PHASE

A **Dummy Phase** is a phase placeholder that you can pull from the Setlist into your graph as empty element when you build a graph on operation level.
To fill a dummy phase with content you have to replace it with another phase from the Setlist.

DUMMY PROCEDURE

A **Dummy Procedure** is a procedure placeholder that you can pull from the Setlist into your graph as empty element when you build a graph on master recipe level.

DUMMY UNIT PROCEDURE

A **Dummy Unit Procedure** is a unit procedure placeholder that you can pull from the Setlist into your graph as empty element when you build a graph on procedure level.

HIDDEN PHASE

A **Hidden Phase** is a structural phase that is required to build adjacent branches or loops in valid SFC syntax (see "What Is an SFC Graph?" in Vol. 1). Thus it is only visible in Recipe and Workflow Designer and does not appear as executable phase during order or workflow processing.

> **TIP**
>
> Please note that the system also provides a **Hidden Phase** that is suitable for use in operations that hold the **server-run** capability (page 109).

Example use cases are:

■ Two branches in direct sequence
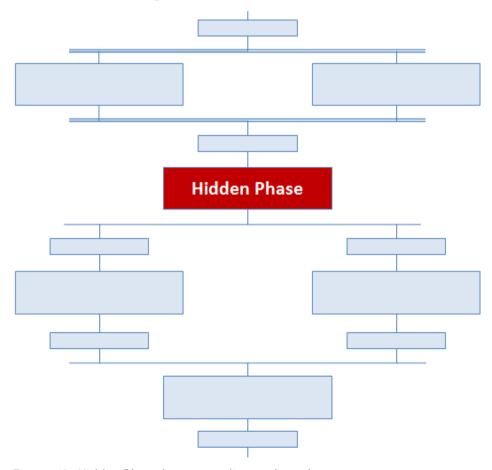
Figure 49: Hidden Phase between adjacent branches

■ A loop to a point directly preceding a branch
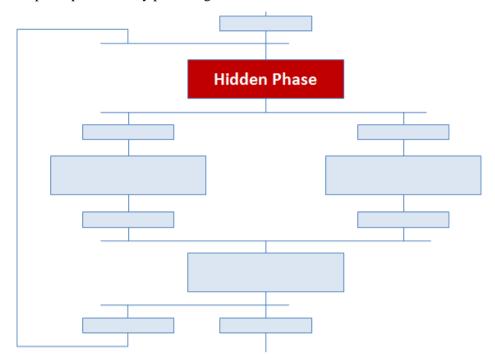
Figure 50: Hidden Phase between loop and branch

MFC Data

Recipe Designer determines the MFC-relevant data of a master recipe or procedure building block from the attributes defined for their material parameters. The **Material Flow Control** tab consists of two panels:

■ The graph panel (page) on the left holds the non-editable MFC graph where you can see the materials defined for your recipe.

■ The table panel (page) on the right holds the non-editable table of MFC items. It lists all MFC items with their material identifiers and indicates where the input materials enter the process.

The panels are divided by a slidable separator and can thus be freely resized, if necessary.
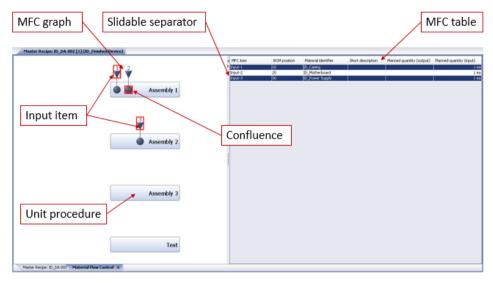
Figure 51: Material Flow Control tab in the Graph Window

MFC Graph Components

The MFC graph displays at which unit procedure the recipe's input materials enter the process. The following external events can affect the MFC graph:

■ changes made in the Parameter Panel, to the **MFC-relevant** or the **Position** attributes of a material parameter,

■ structural changes to the SFC graph, such as the deletion of a phase, operation, or unit procedure with MFC-relevant material parameters.

TIP

Please note that **MFC-relevant** MFC output parameters are not allowed in device recipes.

The MFC graph consists of the following components:

 Unit procedure

The individual unit procedures of the procedure are displayed as horizontal bars and hold all material nodes defined for their respective phases.

 MFC input

An MFC input is displayed as triangle with an outgoing connector and represents a material input parameter configured with one of the procedure's phases. The number it shows corresponds to the numeric suffix of the MFC item identifier, which the system generates and assigns automatically.

When you hover over an MFC input, the system displays the following attributes as tooltip:

Material identifier, **Short description**, **Position** (if available), **Planned quantity** (if available) or **Planned quantity mode** (depending on its mode), and its path within the unit procedure (operation/phase).

Double-click an MFC input to open the Parameter Panel on the procedure level. The system shows the affected material input parameter as selected.

 Confluence

A confluence is displayed as circle and represents the point in the material flow when an input enters the process. So each input corresponds to one confluence.

When you hover over a confluence its tooltip lists the attributes of its input.

Connector

Connector lines represent the material flow between inputs and confluences.

When you hover over a connector, the system displays a tooltip with the **Material identifier** and the **Short description** of the material it represents.

Right-click anywhere on the background to open a shortcut menu with the following function:

- Close sub-tab (CTRL+W)
  Closes the currently active tab of the lower tab bar.

> **TIP**
> Please note that the functions for merging or splitting are not available for device recipes.

MFC Table

The MFC table lists all MFC input items of the master recipe.
You cannot edit the data shown in the table, but selecting or multi-selecting listed items highlights the corresponding components in the MFC graph. Hovering over a table cell displays its content as tooltip. The table updates as a result of the following types of events:

■ changes made in the Parameter Panel, to the **MFC-relevant** or the **Position** attributes of a material parameter,

■ structural changes to the SFC graph, such as the deletion of a phase, operation, or unit procedure with MFC-relevant material parameters.

Double-click a list row in the MFC table to open the Parameter Panel on the procedure level. The system shows the affected material parameters as selected.

The MFC table provides the following data for each of the listed items:

■ **MFC item**
Unique identifier of the item. It consists of the MFC type (input) and a count number.
The identifiers are persistent within the MFC context, which means that when you unselect the **MFC-relevant** option of a material parameter in the Parameter Panel and then re-select it again, it receives a new identifier. The same applies when you delete an MFC-relevant material parameter and re-insert it.

> **TIP**
> Please note that the MFC item identifiers of building blocks are renumbered when you draw the building block into a master recipe.

■ **Position**
You have to define a position number for each MFC-relevant input parameter that is also an MFC input item. Position numbers must be unique throughout each execution path of a master recipe or building block. If you do not define the positions manually or import them from an ERP BOM (page 137), the system indicates in the Messages window (page 57) which input parameters lack their position numbers.
The positions are presented in any list of material inputs, for example during material identification on the shop floor, and are also used as default sorting criterion.

Defined with the material parameter.

■ **Material identifier**
Drawn from the material parameter.

■ **Short description**
Drawn from the material parameter.

■ **Planned quantity (output)**

For output or transfer items of material parameters with **As defined** as planned quantity mode, it shows the planned quantity as defined with the material parameter.

For output or transfer items of material parameters with other planned quantity modes it shows the respective mode, either **None** or **As produced**.

For device recipes, this column is empty, since output parameters are not allowed.

■ **Planned quantity (input)**

For input or transfer items of material parameters with **As defined** as planned quantity mode, it shows the planned quantity as defined with the material parameter.

For input or transfer items of material parameters with other planned quantity modes it shows the respective mode, either **None** or **As produced**.

The list of MFC items is sorted first by type, then by position, and finally by MFC item identifier.

Change Requests for Mass Changes

In a change request, you first select the affected building blocks, the new one you wish to use and the old one you wish to replace. Then you compile the Usage List of the old building block, which lists all of its occurrences. From the list, you can select in which contexts (master recipes or building blocks) the system is to replace the old building block with the new building block and thus create the Action List of the changes to be performed.

IMPORTANT

The **Change request** function is only available to replace an (approved or archived) building block located in either a valid master recipe or an (approved or archived) building block with another approved building block.

For performing the replacement actions, the system uses the **Smart replace** function (page 36). Thus, when configuring the parameters of the new building block, you need to make sure to lock all parameters that are to replace their counterparts on the old building block.
Please note that material parameters are not affected by the **Smart replace** function.

The Change Request tab consists of two panels:

■ The building block panel at the top for the building block images of the new (replacing) and the old (to be replaced) building blocks.
To fill the building block images, proceed as follows:

1. Load the respective building blocks into the Setlist

2. Select the building block image you wish to fill with a building block.

3.   Double-click the respective building block in the Setlist.

This way you can also replace a previously selected building block with another one.
To remove a building block from the change request, select its image and press the DEL key.

■   The table panel underneath holds two sub-tabs:

■   the **Usage List** tab (page 164) to show all occurrences of the old building block and

■   the **Action List** tab (page 166) to show all master recipes and/or building blocks which are to be changed.

The panels are divided by a slidable separator and can thus be freely resized, if necessary.
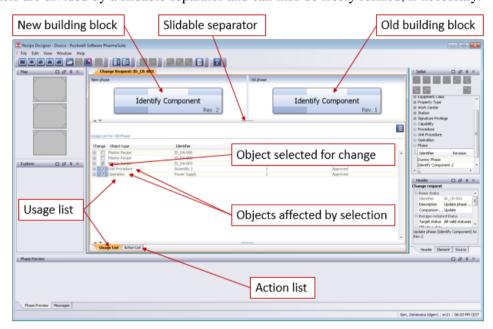
Figure 52: Change Request tab in the Graph Window

Usage List

When you create a new change request or open an existing one, the usage list is empty.

Click the **Compile usage list** button to create the list. The system accesses the database to determine the occurrences of the old building block.

Once you have compiled the usage list, it shows all currently approved occurrences of the building block you have selected as old. In the list, each recipe or building block containing the old building block is displayed as a nested table, with its own data as main table and root node for a collapsible sub-table that lists the data of the individual occurrences.

When listing the individual occurrences of the building block in the sub-tables under the root components, the system differentiates between

- **direct usage** of the building block, when the building block itself has been used as component of the root component's graph, and

- **indirect usage** of the building block, when the building block occurs in the graph because it is located in a higher-level building block, which has been used as component in the root component's graph.

By default, the affected root components are sorted by structure level, but you can adjust both the sort and the column order.

Please note that each time you compile a usage list, the system refreshes the data on which it performs the search. For this purpose it has to access all relevant objects (master recipes and building blocks) on the database to determine if they have been modified since the last time a usage list was compiled.

If there are other users who work with the system on the same database, some objects may be locked as they are currently in use and thus not accessible for refreshing. In this case, the usage list compiles with the potentially obsolete data it has retrieved before its previous run, but lists all objects it could not refresh in the **Locked Objects Unavailable for Refresh** section underneath the usage list.

If the status of an affected object has changed, but the object is locked by another user and thus not available for refresh, it is highlighted in the list of locked objects. You should resolve this issue by recompiling the usage list when the object has been unlocked, as it would prevent the approval of the change request.

Please note that your access rights determine which components are considered when the system compiles the usage list. Confidential recipes or building blocks that you are not allowed to access are not shown in the list.

The panels are divided by a slidable separator and can thus be freely resized, if necessary.

To recompile and thus refresh the usage list, click the **Compile usage list** button.

After you have compiled the usage list you need to define on the basis of the occurrences in which root component you wish to perform the change. Select the **Change** option for each root component whose occurrences of the old building block you wish to replace with the new building block. When you select a root component for the change, the system automatically selects all other components that would also be affected by the planned change because they represent a building block that contains the building block to be replaced as component and thus account for its indirect usage in the root component.
The system indicates the affected building blocks not only by selecting their **Change** options, but also by setting the background color of the cell to blue.

By selecting a root component for changing, you add it to the action list (page 166).
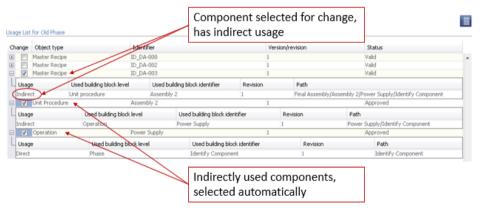
Figure 53: Usage list

Action List

The action list shows all occurrences of the old building block that were selected in the usage list for being included in the change. To add or remove objects you need to return to the **Usage List** tab (page 164) and select or unselect the **Change** option for the respective components.

At its top, the **Action List** tab shows a status and progress indicator with the overall execution status of the change request, which updates with each action performed during the execution of the change request.
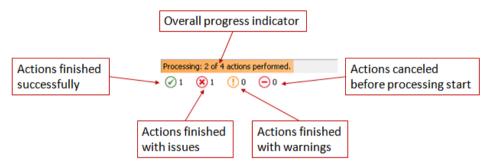
Figure 54: Status and progress indicator

To start the execution of the change request, click the **Execute change request** button. It is only enabled for **Valid** change requests.

TIP

Make sure that all change requests finish their execution prior to their defined effective dates (page 48). A change request that finishes after its effective date would have to change the past and is thus bound to fail.

In order to prevent memory issues that would impact the performance of the system it is recommended that you close large master recipes or building blocks you may have open in the Graph Window.

When performing the actions of a change request, the system uses a bottom-up approach. It starts with changing the building blocks on the lowest available level of the recipe structure and proceeds level by level until it reaches the master recipes. The user interface of Recipe Designer is locked and not available to run other functions while a change request is being executed.

The system visualizes the execution progress for each action in its **Progress** column, with a colored progress bar and a textual display of the seven execution steps to be performed for each action:

1. Preparing changes
   The system accesses the component to lock it and read its current status.

2. Creating new version/revision
   The system performs a **Save as** operation on the component to create the new version/revision on which the change can be performed.

3. Replacing building block
   The system uses the **Smart replace** function (page 36) to perform the intended changes on the building block occurrences to be replaced.

4. Checking consistency
   The system checks the component to make sure it is ready to be **Approved** or moved to **Valid**.

5. Changing status to target status
   The system moves the component to its intended target status. The target status of building blocks is always **Approved** in this step, while the target status of master recipes must be selected during the definition of the change request.

6. Changing status to **Archived**
   This step is only performed on building blocks whose initial revision was **Archived** before the execution of the change request was started.
   The system moves the building block status to **Archived**.

7. Done
   The system has successfully performed all steps of the change request on the component.

TIPS

Make sure that none of the objects affected by a change request are open in your current or another session of Recipe and Workflow Designer while a change request is being executed as this might interfere with changes made to the objects being processed by the change request.

Make sure your system does not move into a hibernation state while a change request is being executed.

If an unexpected system shutdown occurs while a change request is **In process**, you cannot resume the change request after the system has been restarted. You can only manually move it either to **Obsolete** or to **Finished**, depending on what you consider more suitable in view of its progress. Additionally you need to release the locks the change request may still have on the objects it was processing.
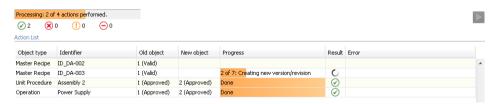
Figure 55: Action list

When an issue occurs that prevents an action from completing successfully, the system provides a short summary of the issue in the action's **Error** column. In the context-sensitive help system, you can find information on how to resolve the issue. To access the relevant help topic, select the issue in the list and press the F1 key.

The following issues can occur:

Locking issue: cannot open the component to be changed.

Make sure the object is not locked any more, create a new change request for the component, and execute it.

Status issue: status of the component changed after approval of the change request.

Create a new change request for the component and execute it.

Version issue: version of the component changed after approval of the change request.

Create a new change request for the component and execute it.

Component creation issue: cannot save the new version/revision of the component.

This is a rare issue that can only occur if the number the system has calculated for the new version/revision of the component has in the meantime been used by someone else for creating another version/revision of the component.
Create a new change request for the component and execute it.

Indirect usage replacement issue: cannot find the new revision of the replacing building block.

The system could not create the new revision of the building block that is required to perform the replace action on the component. This can happen when the system fails to create the building block that is used indirectly in the building block of the current component, e.g. on account of a locking issue.
Make sure the indirect usage can be processed, create a new change request for the component, which automatically includes the building blocks used indirectly, and execute it.

Parameter replacement issue: cannot merge parameters of new building block into old building block.

Create a new change request that holds a building block with compatible parameters as new building block and execute it.

Consistency issue: new version/revision of the component has unresolved issues listed in its Messages window.

Open the component in the Graph Window and resolve the issues.

Status data issue: cannot set the effective date of the recipe.

Create a new change request with a valid effective date for the component and execute it.

Internal error.

An issue of this type can be caused by network instability disrupting the communication with the database. Create a new change request for the component and try again or contact your system administrator.